



TREBALL DE FI DE GRAU

GRAU EN ENGINYERIA ELECTRÒNICA DE TELECOMUNICACIÓ

ESTABLIMENT D'UN CANAL DE COMUNICACIONS EN LÍQUID MITJANÇANT PMUTs

Monforte Gras, Marc

DIRECTOR: Núria Barniol Beumala

DEPARTAMENT D'ENGINYERIA ELECTRÒNICA

UNIVERSITAT AUTÒNOMA DE BARCELONA

Bellaterra, Setembre 2020

Resum

RESUM: Aquest treball busca estudiar les possibilitats de formar un canal de comunicacions inalàmbriques, mitjançant ultrasons, amb dispositius del tipus PMUT. En aquest sentit, s'implementen simulacions realitzades conjuntament en els programes Matlab i LTspice. Amb Matlab s'empleen llibreries com el FIELD II per simular camps ultrasònics originats per transductors, i s'estableix un enllaç amb LTspice per poder simular la resposta obtinguda al passar pels filtres freqüencials necessaris per una comunicació Full-Duplex. Finalment, s'utilitzarà el protocol BPSK per establir el sistema de comunicacions.

RESUMEN: Este trabajo busca estudiar las posibilidades de formar un canal de comunicaciones inalámbricas, mediante ultrasonidos, con dispositivos del tipo PMUT. En este sentido, se implementan simulaciones realizadas conjuntamente en los programas Matlab y LTspice. Con Matlab se usan librerías como FIELD II para simular campos ultrasónicos originados por transductores, y se establece un enlace con LTspice para poder simular la respuesta obtenida al pasar por los filtros frecuenciales necesarios para una comunicación Full-Duplex. Finalmente, se usará el sistema BPSK para establecer el sistema de comunicaciones.

SUMMARY: This dissertation looks for studying the possibilities of forming a wireless communication channel, through ultrasounds use, with PMUT devices. To do that, simulations are made jointly in Matlab and LTspice programs. With Matlab libraries like FIELD II are used to simulate ultrasonic fields originated by transducers, and a link is established with LTspice in order to be able to simulate the response obtained when passing through frequency filters needed for a Full-Duplex communication. Finally, a BPSK system will be used to form the communication system.

Presentació

Degut a la pandèmia del COVID-19 aquest treball s'ha hagut d'adaptar als nous protocols i mesures prohibitives, com el tancament dels laboratoris. Per aquest motiu l'abast i objectius inicials s'han vist modificats, i el treball esdevindrà efectuat des d'un punt de vista purament teòric i simulat.

Donada la necessitat global de desenvolupar tecnologies amb millores en miniaturització, precisió i baix consum d'energia entre d'altres, els materials piezoelèctrics esdevenen un component essencial en la recerca de nous dispositius. Això és degut a la seva naturalesa i el mecanisme de fabricació, que permeten estructures de tipus MEMS (*Microelectromechanical Systems*) amb nivells molt baixos d'alimentació.

Actualment, és possible la deposició de materials piezoelèctrics a baixes temperatures mitjançant tecnologia de fotolitografia, que els fa compatibles amb el seu processat en oblies de silici en les que prèviament s'hi han fabricat circuits CMOS. D'aquesta manera es pot assolir la màxima miniaturització a preus accessibles per grups de recerca.

Per confirmar la possibilitat d'usar PMUTs (*Piezoelectric Micromachined Ultrasonic Transducers*) com a transceptors en un sistema Full-Duplex de telecomunicacions ultrasònic en medi líquid és necessari ser capaços de simular tots els elements que el formen, i d'establir diferents valors de SNR (*Signal-to-Noise Ratio*) per veure com afecta al BER (*Bit Error Rate*) del sistema.

Es seguirà un cert número de passos per tal de realitzar la simulació esmentada:

1. Creació dels senyals i el soroll a transmetre a Matlab.
2. Ús del programari FIELD II per simular la transmissió del medi i trobar els senyals rebuts als receptors.
3. Enviament del senyal filtrat a LTspice per separar els senyals mitjançant les seves freqüències.
4. Reenviament dels senyals ja separats a Matlab per calcular el BER en funció del SNR.

Índex

Resum	i
Presentació	iii
Llista de Figures	ix
Llista de Taules	xi
1 Introducció	5
1.1 PMUT	5
1.1.1 Efecte piezoelèctric	7
1.2 Canal de Comunicacions	8
1.2.1 Medi de treball	9
1.2.2 BPSK	9
1.2.3 BER	10
1.2.4 SNR	11
2 Treballar amb Matlab i FIELD II	13
2.1 Formació del transductor	13
2.2 Formació del canal	16
3 Generació del filtre amb LTspice	19
3.1 Primer filtre, a 4MHz	21

3.2	Segon filtre, a 6 MHz	22
3.3	Unió dels filtres generats	23
4	Enllaçament dels senyals generats a Matlab i LTspice	27
4.1	Esquema de funcionament de l'interfície LTspice-Matlab	27
4.2	Transmissió de dades de Matlab a LTspice	28
4.3	Transmissió de dades de LTspice a Matlab	31
5	Comprovacions del correcte funcionament del sistema de telecomunicacions.	35
5.1	Elaboració del missatge a emetre amb Matlab i FIELD II	35
6	Simulacions i anàlisis del sistema de telecomunicacions. Comprovació del BER en funció del SNR	45
6.1	Generació i captació d'un missatge aleatori sense soroll	45
6.1.1	Generació dels bits d'entrada i del senyal a emetre. Simulació amb FIELD II	46
6.1.2	Filtratge del senyal amb LTspice	49
6.1.3	Captació de la informació de LTspice i obtenció dels bits de sortida	49
6.1.4	Càlcul del BER per les dues freqüències	52
6.2	Modificació del BER obtingut en funció del valor de SNR afegit al sistema. . . .	52
	Conclusions	59
	Referències	i
	Referències a articles	ii
	Referències a procediments	iii
	Referències a pàgines web	iii

Índex de figures

1.1	Vistes en secció i alçat d'un PMUT.	5
1.2	Esquema de la comunicació Full-Duplex que serà utilitzada al projecte.	7
1.3	a) Efecte piezoelèctric directe. b) Efecte piezoelèctric invers.	7
1.4	Elements d'un canal de comunicació.	8
1.5	De dalt a baix, senyals modulada, moduladora i portadora.	9
2.1	Inicialització del programari FIELD II.	13
2.2	Codi empleat per establir el transductor desitjat.	14
2.3	Codi empleat per visualitzar el transductor desitjat.	14
2.4	Vista tridimensional de l'array de 10x10 PMUTs.	15
2.5	Vista en planta de l'array de 10x10 PMUTs, amb millor visualització de dimensions.	15
2.6	Codi generat per realitzar i visualitzar un estímul de dues longituds d'ona.	16
2.7	Condicions de treball de FIELD II per cada iteració freqüencial.	17
2.8	Gràfiques generades per les dues freqüències on es visualitzen els senyals captats pels transductors.	18
2.9	Figura 2.8 amb cursors per evaluar la freqüència obtinguda.	18
3.1	Estructura general d'un filtre de 3r ordre passa-baixos.	20
3.2	Transformació dels elements d'un filtre passa-baixos a un passa-banda.	20
3.3	Filtre passa-banda general.	21
3.4	Filtre passa-banda de 3r ordre a 4MHz amb aproximació Butterworth.	21
3.5	Paràmetres S_{11} i S_{21} del filtre 3.4.	22

3.6	Filtre passa-banda de 3r ordre a 6MHz amb aproximació Butterworth.	23
3.7	Paràmetres S_{11} i S_{21} del filtre 3.6.	23
3.8	Filtre resultant de la combinació dels filtres 3.4 i 3.6.	24
3.9	Simulació del filtre 3.8 amb cursors a les freqüències de tall del primer filtre. . . .	25
3.10	Simulació del filtre 3.8 amb cursors a les freqüències de tall del segon filtre. . . .	25
3.11	Augment a la banda passant de la simulació del filtre 3.8 amb cursors a les freqüències més atenuades.	26
4.1	Esquema del tractament de dades general per simular el sistema de telecomunicacions.	28
4.2	Codi utilitzat per generar arxius <i>.txt</i> que LTspice utilitzarà per generar tensions per PWL.	29
4.3	Prova de la funció <i>crearPWL</i> , genera els arxius <i>test1.txt</i> i <i>test2.txt</i>	29
4.4	Plotting de les sortides generades per Matlab, <i>out1</i> i <i>out2</i> , per comparar amb les senyals captades a LTspice.	30
4.5	Esquemàtic per simular els arxius generats a Matlab.	30
4.6	Simulació dels arxius PWL <i>test1</i> , en vermell, i <i>test2</i> , en blau.	31
4.7	Dades generades amb l'ús de la funció <i>LTspice2Matlab</i>	31
4.8	Contingut de les variables <i>variable_mat</i> i <i>variable_name_list</i> i <i>time_vect</i>	32
4.9	Gràfic de la tensió V_{out1} obtinguda de LTspice.	32
5.1	Codi utilitzat per generar els senyals a partir de la cadena de bits a enviar. . . .	36
5.2	Gràfic de l'excitació generada enfront del senyal rebut pel PMUT.	37
5.3	Senyals generats a 4, 6 MHz i la suma d'ambdós, respectivament.	37
5.4	Filtre modificat per tenir com a entrada un senyal i sortides a les dues freqüències, 4 i 6 MHz.	38
5.5	Sortida del filtre a 4MHz.	39
5.6	Sortida del filtre a 6MHz.	39
5.7	Entrada del filtre generada amb l'arxiu PWL.	40
5.8	Sortida del filtre a 4MHz comparada amb un senyal sinusoidal pur a 4MHz. . . .	41

5.9	Sortida del filtre a 6MHz comparada amb un senyal sinusoidal pur a 6MHz. . . .	41
5.10	Dades provinents de la simulació en LTspice i tractades amb Matlab.	42
5.11	Demostració de la no-linealitat de les mostres temporals.	42
5.12	Codi generat per detectar els bits de sortida.	44
6.1	Establiment dels paràmetres generals i creació del missatge.	46
6.2	Generació de l'excitació del PMUT per cada una de les freqüències.	46
6.3	Vector d'excitació combinat per les dues freqüències.	47
6.4	Establiment dels paràmetres de FIELD II i creació del PMUT.	47
6.5	Aplicació al PMUT de la resposta impulsional típica i l'excitació calculada. . . .	47
6.6	Establiment de la distància de comunicació i simulació del senyal resultant. . . .	48
6.7	Alliberament de memòria i generació de l'arxiu PWL.	48
6.8	Filtre utilitzat a LTspice per separar les freqüències del senyal rebut.	49
6.9	Codi generat per extreure els senyals separats freqüencialment i el seu vector temps.	50
6.10	Codi generat per detectar els bits de sortida del senyal a 4MHz.	50
6.11	Codi generat per detectar els bits de sortida del senyal a 6MHz.	51
6.12	Comprovació del BER per cada freqüència treballada.	52
6.13	Línies utilitzades per incorporar el valor de SNR desitjat al sistema.	53
6.14	Excitacions abans (blau) i després (vermell) del soroll per diferents valors de SNR. Freqüència a 4 MHz	54
6.15	Excitacions abans (blau) i després (vermell) del soroll per diferents valors de SNR. Freqüència a 6 MHz	55

Índex de taules

3.1	Valors per generar el primer filtre passa-banda	21
3.2	Valors per generar el segon filtre passa-banda	22
3.3	Valors per generar el filtre passa-banda complert.	24
6.1	Obtenció dels diferents BER del sistema per cada SNR i freqüència.	56

Motivació i objectiu

Aquest projecte pretén establir els límits màxims de soroll que pot assolir un sistema de comunicacions Full-Duplex on els transceptors són dispositius piezoelèctrics sumergits en líquid.

Amb aquest objectiu s'intenta obtenir un límit, mitjançant simulacions, del soroll màxim que pot arribar a tolerar un sistema amb les característiques esmentades. Això és degut a que els elements piezoelèctrics en sistemes de comunicació conformen un camp científicotecnològic en actual expansió i investigació donades les seves característiques de baix consum i capacitat de miniaturització.

El treball està motivat per les grans possibilitats aplicatives dels sistemes esmentats. Al ser dispositius de mida tan reduïda i baix consum, es poden implementar com a sensors biomètrics de llarga durada. Només en aquest àmbit es permetria una millora substancial en confort i adquisició d'informació mèdica, a més d'un feedback en temps real que els sensors poden proporcionar sobre l'estat de salut de la persona que els porta. Adicionalment, però no menys important, els ultrasons ja són àmpliament emprats com a eina de diagnòstic donat que es tracta d'una tecnologia no invasiva.

Més enllà de la medicina i altres aplicacions amb necessitat de biometria, aquests sistemes es poden utilitzar en qualsevol comunicació subaquàtica de curta distància, mantenint sempre un perfil baix d'alimentació i mida.

Estat de l'art

La piezoelectricitat és un fenomen conegut des de l'any 1880 per Pierre i Jacques Curie. De manera molt breu, hi ha certs materials que sota una certa deformació física generen una diferència de voltatge, i en ser sotmesos a una caiguda de tensió pateixen una deformació física.

Es va trigar més de 30 anys en obtenir una aplicació realment important. Aquesta va ser obtinguda l'any 1917 per Paul Langevin en conjunt amb altres científics francesos. Consistia en un transductor piezoelèctric que es situava sota l'aigua, emitia i rebia un pols. Amb el temps entre emissió i recepció es podia conèixer la profunditat de l'aigua¹.

És destacable observar que l'essència del projecte actual segueix els mateixos passos que varen descobrir els científics durant la primera guerra mundial. És clar que en el seu cas es treballa en detectar l'existència d'un únic pols a 50 Hz i ara es busquen transductors molt més petits, amb necessitats d'alimentació reduïdes i freqüències de treball de l'ordre dels MHz.

Tot i que els piezoelèctrics es van començar a utilitzar en elements de telecomunicació a partir del 1920 aproximadament, només es podien obtenir bons resultats per cable. No sent fins fa poc temps que es va iniciar la implantació de sistemes de comunicació a distància sense fils mitjançant l'efecte piezoelèctric.

Actualment sí es poden trobar articles recents que inicien investigacions en aquest àmbit, començant a produir-se més freqüentment a partir del 2015.

Aquests articles es mencionen al respecte en diferents aspectes de l'àmbit piezoelèctric, com poden ser:

- Comparació de la tecnologia de comunicació piezoelèctrica amb altres ja existents².
- Generació de dispositius funcionals amb la tecnologia esmentada³.

¹Midé Technology Engineering Corporation. *The History of Piezoelectricity*. URL: <https://piezo.com/pages/history-of-piezoelectricity> (cons. 28-08-2020).

²Adam Gannon et al. "Wireless Communications in Intra-Body Networks". A: (2016).

³Flavius Pop et al. "Novel pMUT-Based Acoustic Duplexer for Underwater and Intrabody Communication". A: Oct. de 2018, pàg. 1-4. DOI: 10.1109/ULTSYM.2018.8579709.

- Establiment de xarxes de comunicació intracorpòries mitjançant ultrasons^{4,5}.
- Monitorització i control mèdic del cos en temps real⁶.
- Recerca i millora dels actuals dispositius piezoelèctrics^{7,8}.

Aquesta densitat d'articles científics referents al tema que es tracta demostra que és un camp en expansió i amb un potencial tecnològic amb un abast encara no ben definit.

⁴G. Enrico Santagati i Tommaso Melodia. "Experimental Evaluation of Impulsive Ultrasonic Intra-Body Communications for Implantable Biomedical Devices". A: *IEEE Transactions on Mobile Computing* 16 (gen. de 2016), pàg. 1-1. DOI: 10.1109/TMC.2016.2561277.

⁵Bernard Herrera et al. "Frequency Agile PMUT-Based Ultrasonic Communication Links". A: Gen. de 2019, pàg. 974-977. DOI: 10.1109/MEMSYS.2019.8870840.

⁶Flavius Pop et al. "Enabling Real-Time Monitoring of Intrabody Networks through the Acoustic Discovery Architecture". A: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* PP (juny de 2020), pàg. 1-1. DOI: 10.1109/TUFFC.2020.3002973.

⁷Eyglis Ledesma et al. "Tent-plate AlN PMUT with a piston-like shape under liquid operation". A: *IEEE Sensors Journal* PP (maig de 2020), pàg. 1-1. DOI: 10.1109/JSEN.2020.2995643.

⁸Tao Wang, Renshi Sawada i Chengkuo Lee. "A Piezoelectric Micromachined Ultrasonic Transducer (pMUT) Using Piston-like Membrane Motion". A: *Electron Device Letters, IEEE* 36 (set. de 2015), pàg. 957-959. DOI: 10.1109/LED.2015.2459075.

Capítol 1

Introducció

En aquest capítol s'establiran els coneixements previs necessaris i els principis físics de funcionament d'aquest treball. S'explicarà l'estructura dels PMUTs i la seva funció com a transceptor (emissor i receptor simultàniament) en un canal de comunicacions. A més, es detallarà la seva incorporació en un sistema de comunicacions Full-Duplex.

1.1 PMUT

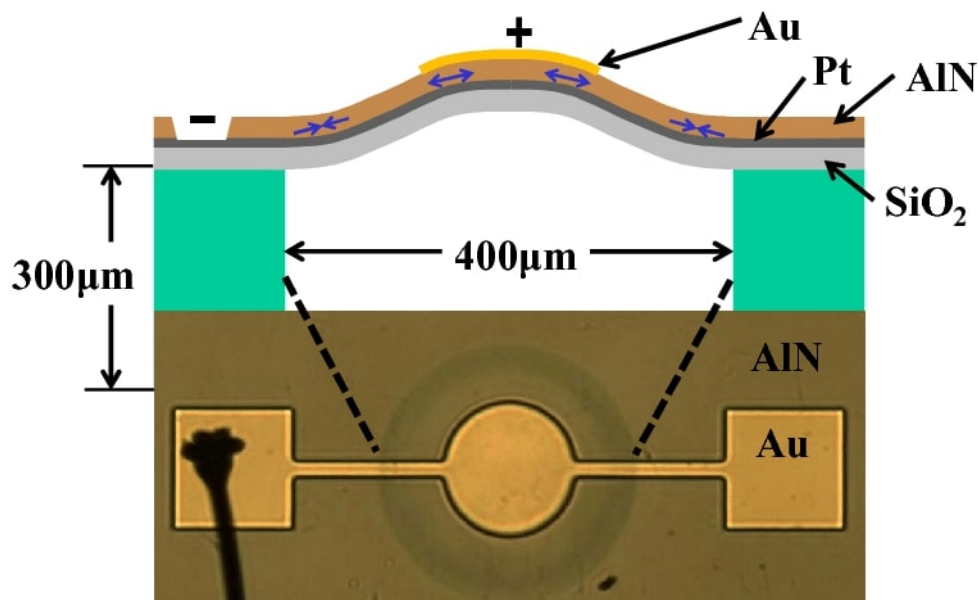


Figura 1.1: Vistes en secció i alçat d'un PMUT.

Les sigles PMUT provenen de l'anglès *Piezoelectric Micromachined Ultrasonic Transducer*. Aquests elements es caracteritzen per una mida petita, de l'ordre de les dècimes o centèsimes

de mil·límetre. Per aquest mateix motiu, la seva banda freqüencial més aprofitable és la dels ultrasons (superiors a 20KHz, fins a l'ordre dels GHz).

Com es visualitza a 1.1, un PMUT es compon d'una membrana piezoelèctrica ubicada de manera que pugui flexar sota l'influència d'una ona de pressió acústica i, per la seva naturalesa, produir una diferència de potencial que es pot mesurar i està directament relacionat amb el desplaçament de la membrana respecte el seu estat de repòs¹.

De la mateixa manera, exposat a una diferència de voltatge, un PMUT flexarà la seva membrana en relació al potencial aplicat. Aquests dos fenòmens provoquen que siguin dispositius aptes per rebre o emetre, respectivament, ones mecàniques.

De tota manera, en funció de la mida i geometria del transductor s'estableixen modes de ressonància a diferents freqüències. És necessari utilitzar el primer mode de ressonància, donat que és el mode amb més moviment i que generarà una ona de pressió acústica més fàcilment mesurable.

Per poder emetre i rebre simultàniament es realitzarà una multiplexació freqüencial. Ambdós senyals tindran freqüències suficientment diferenciades com per poder ser separades amb un filtre passabanda de tercer ordre.

Cal tindre en compte que els PMUTs són idèntics, per tant també ho serà la seva freqüència de ressonància principal. Per aquest motiu no es poden establir freqüències de treball arbitràriament distanciades. Es treballarà amb freqüències de 4 i 6 MHz, establint que la freqüència del primer mode de ressonància es troba a 5 MHz. D'aquesta manera ambdós PMUTs treballen a 1MHz de la seva freqüència ideal.

Per efectuar la transmissió Full-Duplex entre dos transductors és necessari que ambdós puguin emetre i rebre alhora. Per tal efecte, mentre el PMUT "A"emet a 4MHz i rep a 6MHz, el transductor "B"emet a 6MHz i rep a 4MHz. A efectes de simulació, donat que es treballa amb dos transceptors idèntics, el programa serà transparent al concepte de PMUTs "A" i "B", però és necessari saber que s'estableixen dos canals de comunicació, un en cada sentit de la transmissió d'informació.

És important senyalar que degut al petit tamany dels PMUTs i els voltatges que s'empleen, es treballa amb arrays dels mateixos per poder obtenir pressions acústiques superiors. En el nostre cas simularem un array de 10x10 PMUTs.

¹Junhong Li et al. "Design and Fabrication of Piezoelectric Micromachined Ultrasound Transducer (pMUT) with Partially-Etched ZnO Film". A: *Sensors (Switzerland)* 17 (juny de 2017). DOI: 10.3390/s17061381.

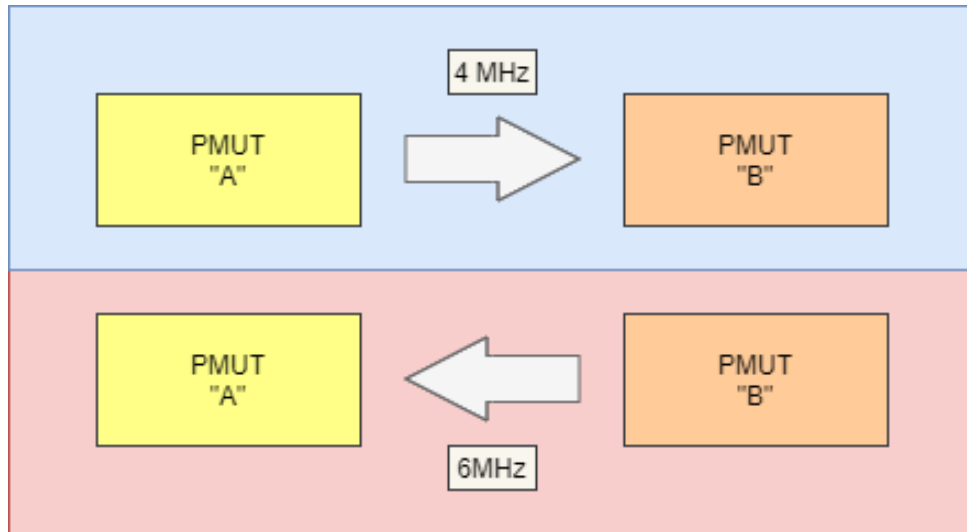


Figura 1.2: Esquema de la comunicació Full-Duplex que serà utilitzada al projecte.

1.1.1 Efecte piezoelèctric

L'efecte piezoelèctric es coneix en alguns materials cristal·lins des de l'any 1880, de la mà de Pierre i Jacques Curie.

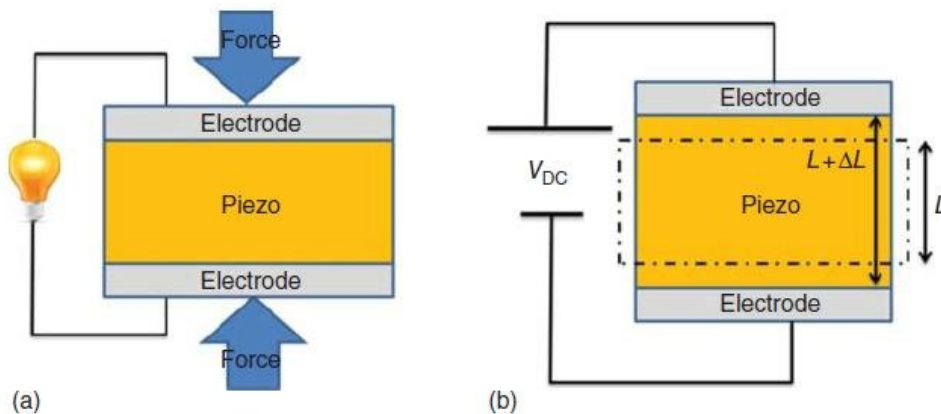


Figura 1.3: a) Efecte piezoelèctric directe. b) Efecte piezoelèctric invers.

El seu funcionament radica en les celdes de l'estructura, que contenen àtoms ionitzats. Quan aquestes celdes es desplacen, es produeix un moviment de les càrregues, que genera una diferència de voltatge mesurable, és l'anomenat efecte piezoelèctric directe.²

En sentit contrari, al sotmetre el cos a una diferència de potencial, les celdes es veuen afec-

²Carmen Emily Yang. *What is the Piezoelectric Effect?* Set. de 2016. URL: <https://www.electronicdesign.com/power-management/article/21801833/what-is-the-piezoelectric-effect#:~:text=%5C%20What%5C%20is%5C%20the%5C%20Piezoelectric%5C%20Effect%5C%3F%5C%20%5C%201,inverse%5C%20piezoelectric%5C%20effect%5C%20sends%5C%20out%5C%20a...%5C%20More%5C%20> (cons. 22-08-2020).

tades per les forces electrostàtiques i generen la deformació del material, és l'anomenat efecte piezoelèctric invers.

1.2 Canal de Comunicacions

Tot sistema de comunicacions ha de tenir una sèrie d'elements, els quals es poden llistar segons la seva funció:

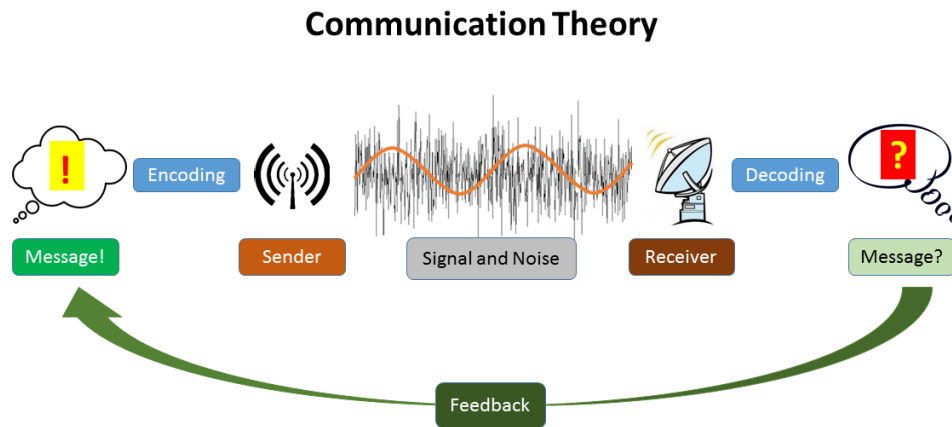


Figura 1.4: Elements d'un canal de comunicació.

- **Emissor:** S'encarrega d'enviar el missatge a través del canal. És directament responsable d'aportar una potència suficient per tal que la informació arribi al receptor.
- **Receptor:** Té la funció de captar el missatge, prèviament enviat per l'emissor.
- **Missatge:** És la informació que es vol transmetre.
- **Canal:** Mecanisme emprat per transportar el missatge entre l'emissor i el receptor. Generalment el canal és el medi o medis que connecten físicament emissors i receptors.
- **Protocol:** Llenguatge o codi emprat per emmagatzemar la informació.

En realitzar una analogia amb el treball actual, es veu que tant l'emissor com el receptor són els PMUTs, el missatge és arbitrari, el canal és (a efectes pràctics) similar a l'aigua, ja que en líquids la velocitat del so és molt més elevada que a l'aire, i el protocol empleat serà el BPSK (*Binary Phase Shift Keying*), ja que té un bon comportament en entorns sorollosos.

1.2.1 Medi de treball

L'entorn on treballaran els transceptors és de vital importància, donat que modifica, entre altres factors, la velocitat de l'ona i l'atenuació que rebrà del medi.

En el nostre cas, es simularà un medi aquàtic, donades les característiques de semblança amb l'interior del cos humà en factors com la velocitat del so i l'atenuació que rebrà.

Els senyals acústics es mouen a 1500m/s en aquest medi, i pateixen una atenuació de 1dB/mm³.

1.2.2 BPSK

El sistema de modulació BPSK es basa en tenir una senyal portadora de freqüència i amplitud fixes. La seva fase es veurà modificada en 180° cada cop que es produeixi un canvi de valor en el bit que es vol transmetre (1 i 0). Aquest mecanisme assoleix BER menors en entorns sorollosos, ajudant a generar un sistema robust⁴.

Aquesta major immunitat al soroll respecte altres modulacions com pot ser la ASK (*Amplitude Shift Keying*) es deu a que la captació del bit es realitza en un màxim o un mínim d'amplitud (exemple: 1V o -1V) mentre que que ASK es capta en un màxim o en ausència de tensió (exemple: 1V o 0V).

Per tant, treballant amb la mateixa amplitud, amb BPSK es pot treballar amb el doble d'error en front d'ASK, per exemple.

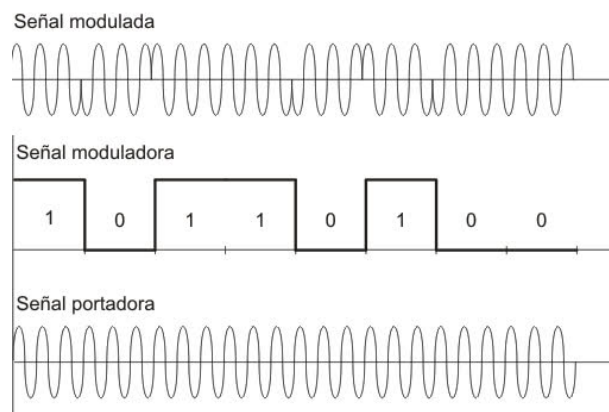


Figura 1.5: De dalt a baix, senyals modulada, moduladora i portadora.

Amb aquest tipus de modulació es calcularà el seu BER (Bit Error Rate) en funció del soroll

³G. Enrico Santagati i Tommaso Melodia. "Experimental Evaluation of Impulsive Ultrasonic Intra-Body Communications for Implantable Biomedical Devices". A: *IEEE Transactions on Mobile Computing* 16 (gen. de 2016), pàg. 1-1. DOI: 10.1109/TMC.2016.2561277.

⁴RF Wireless World. *BPSK-Binary Phase Shift Keying*. 2012. URL: <https://www.rfwireless-world.com/Terminology/BPSK.html> (cons. 22-08-2020).

(SNR, Signal to Noise Ratio) que pot establir-se al canal.

Es prefereix aquesta modulació en front d'altres que funcionin per amplitud (AM) perquè permet treballar en entorns sorollosos. En modulació per amplitud amb un entorn sorollós els 0s poden saltar fàcilment a captar-se com a 1s.

Amb la modulació BPSK escollida els bits sempre tenen valor, per aquest motiu es tracta d'una modulació més adient al propòsit del treball.

1.2.3 BER

El BER és un paràmetre important en els canals de comunicació digital. Donat que en la recepció d'informació sempre hi ha bits que es reben amb valor incorrecte, aquest paràmetre determina cada quants bits un és incorrecte⁵.

Donada la seva definició, la fórmula que en regeix el valor és la següent:

$$BER = \frac{\text{Bits erronis}}{\text{Bits totals}}$$

Segons el criteri utilitzat a l'article [Pop+18]⁶, un BER de valor menor a 10^{-4} és suficient per retransmetre senyals de vídeo en streaming. Per aquest motiu, aquest BER serà suficient per realitzar comunicacions senzilles i contrastades.

Hi ha múltiples factors que poden afectar el BER, alguns principals són:

- Interferències
- Potència del senyal emés
- Ample de banda

En aquest cas s'obtindrà un BER en condicions prou ideals, donat que no patirem interferències ni velocitats de grup anòmales (senyals o trossos del senyal que es generen més tard però arriben abans al receptor). El senyal rebut és molt semblant en forma al senyal emés.

Per altra banda, a més potència de senyal menor SNR, per tant el senyal serà més clar i fàcil de captar.

Succeeix un efecte similar si es redueix l'ample de banda. La potència de soroll captada és menor, el que redueix el SNR amb resultat similar.

⁵Electronics Notes. *What is Bit Error Rate: BER tutorial*. 2016. URL: <https://www.electronics-notes.com/articles/radio/bit-error-rate-ber/what-is-ber-definition-tutorial.php> (cons. 22-08-2020).

⁶Flavius Pop et al. "Novel pMUT-Based Acoustic Duplexer for Underwater and Intrabody Communication". A.: Oct. de 2018, pàg. 1-4. DOI: 10.1109/ULTSYM.2018.8579709.

1.2.4 SNR

El SNR constata la relació entre la potència del senyal i la del soroll del medi⁷.

Donat que amb els PMUTs es treballa amb pressions d'ordres baixos, és críticament necessari que el sistema pugui treballar amb valors de SNR baixos. Això és degut a que es treballa amb valors d'ordre propers al soroll de fons, pel que el SNR mínim ha de ser el més baix possible per tal que es capti el senyal encara que tingui una potència similar al soroll.

⁷Cadence PCB Solutions. *What is Signal to Noise Ratio and How to calculate it?* Gen. de 2020. URL: <https://resources.pcb.cadence.com/blog/2020-what-is-signal-to-noise-ratio-and-how-to-calculate-it> (cons. 22-08-2020).

Capítol 2

Treballar amb Matlab i FIELD II

En aquest capítol es farà èmfasi en la programació necessària per realitzar les simulacions amb el programari FIELD II. Es treballa amb FIELD II mitjançant Matlab.

2.1 Formació del transductor

Per tal d'iniciar els càlculs i les simulacions, el primer pas consisteix en recrear el conjunt de PMUTs amb que es pretenia treballar. En el nostre cas, es tractava d'una configuració d'array quadrada (tant l'array com els PMUTs), amb 10 x 10 elements de longitud lateral 40um i separació de 20um.

En primera instància s'executa la línia "field_init" per inicialitzar el FIELD II. A continuació, es genera el següent codi per establir el transductor desitjat:

```
>> field_init
*-----*
*                                     *
*               F I E L D   I I      *
*                                     *
*           Simulator for ultrasound systems      *
*                                     *
*           Copyright by Joergen Arendt Jensen      *
*   Version 3.24a, May 24, 2014 (Matlab 8.20 gcc version)  *
*           Web-site: field-ii.dk                  *
*                                     *
*   This is citationware. Note the terms and conditions      *
*   for use on the web-site at:                        *
*           field-ii.dk/?copyright.html              *
*   It is illegal to use this program, if the rules in the      *
*   copyright statement is not followed.              *
*-----*
```

Figura 2.1: Inicialització del programari FIELD II.

```

function Th = Square_Transducer()

%Creacio d'un transductor quadrat

no_elem_x = 10;    %nombre d'elements a l'eix X
no_elem_y = 10;    %nombre d'elements a l'eix Y
width = 40e-6;     %amplada dels elements
height = 40e-6;    %alçada dels elements
kerf_x = 20e-6;    %distancia entre elements a l'eix X
kerf_y = 20e-6;    %distancia entre elements a l'eix Y
focus = [0,0,1e-3];
enabled = ones(no_elem_x, no_elem_y);    %matriu que indica quins elements s'activen

Th = xdc_2d_array(no_elem_x, no_elem_y, width, height, kerf_x, kerf_y, enabled, 1, 1, focus);

plotter(Th)

end

```

Figura 2.2: Codi empleat per establir el transductor desitjat.

Per confirmar si els transductors s'han establert de la manera desitjada, s'emplea la funció visible al final del codi, "plotter".

```

function res = plotter(Th)

% Aquesta funció realitza un plot de la superfície del transductor

colormap(copper)

data = xdc_get(Th, 'rect');
[N,M] = size(data);

for i=1:M
    x=[data(11,i), data(20,i); data(14,i), data(17,i)]*1000;
    y=[data(12,i), data(21,i); data(15,i), data(18,i)]*1000;
    z=[data(13,i), data(22,i); data(16,i), data(19,i)]*1000;
    c=data(5,i)*ones(2,2);
    hold on
    surf(x,y,z,c)
end

view(3)
xlabel('x [mm]')
ylabel('y [mm]')
zlabel('z [mm]')
grid
hold off

end

```

Figura 2.3: Codi empleat per visualitzar el transductor desitjat.

Com es veu a 2.3, s'investiguen les formes rectangulars que componen el transductor i es grafiquen en un eix tridimensional.

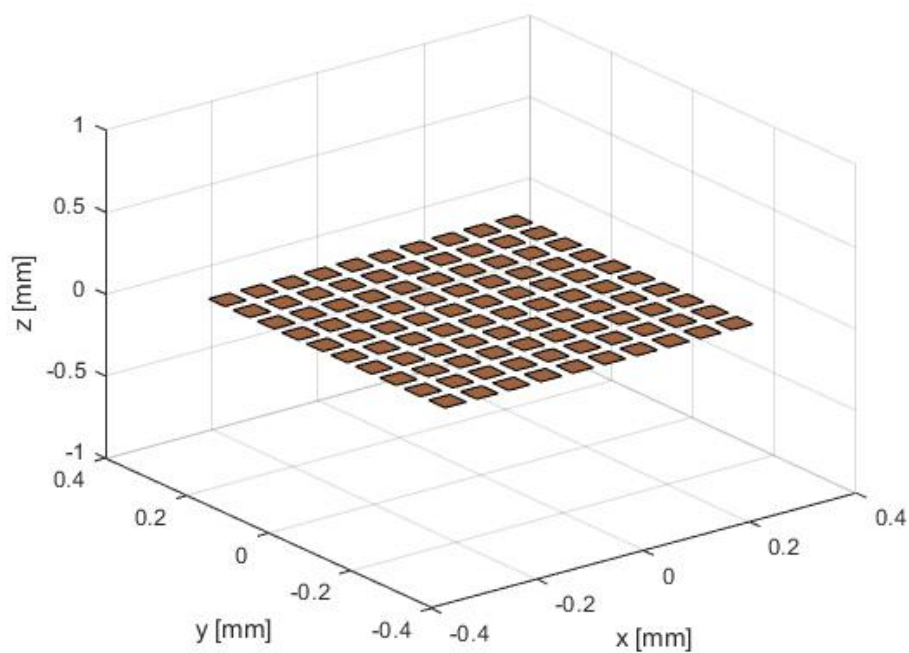


Figura 2.4: Vista tridimensional de l'array de 10x10 PMUTs.

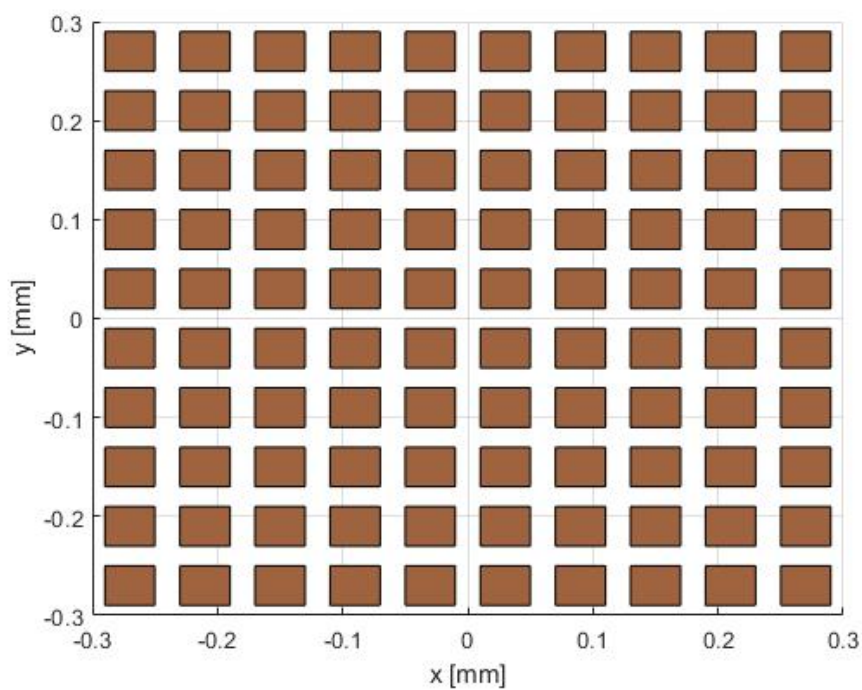


Figura 2.5: Vista en planta de l'array de 10x10 PMUTs, amb millor visualització de dimensions.

Com es contempla a 2.4 i 2.5, l'array s'ha format correctament amb les mesures desitjades. 40um de costat per PMUT i 20 um de separació entre elements, en formació quadrada.

2.2 Formació del canal

Per efectuar la transmissió del missatge cal usar FIELD II per transmetre i rebre el missatge estipulat. Amb aquest fi, s'inicia el procés enviant un pols de dos períodes i obtenint el càlcul del missatge rebut.

```

1 - close all
2 - clear all
3
4 - hold on
5 - cont = 0;
6
7 - freq = [4, 6]*1e6;      %frecuencies centrals
8 - set_field('c', 1500);  %velocitat del so [m/s]
9 - set_field('att', 1);   %atenuació del medi
10 - set_field('use_att',1); %Imposem us d'atenuacio
11 - Th = Square_Transducer(); %apertura d'emissió i recepció
12
13 - for i = freq            %Bucle per calcular les dues frecuencies centrals de treball
14
15 -     f0 = i;            %Frecuencia central
16 -     fs = 100*f0;       %Frecuencia de mostreig, per assegurar un bon càlcul
17 -     set_sampling(fs);  %Establiment de fs a FIELD II
18
19 -     field_info          %Comprovacio que el medi es troba en l'estat desitjat
20
21 -     impulse_response = sin(2*pi*f0*(0:1/fs:2/f0));
22 -     impulse_response = impulse_response.*hanning(max(size(impulse_response)));
23 -     xdc_impulse(Th,impulse_response)
24
25 -     excitation = sin(2*pi*f0*(0:1/fs:2/f0)); %2 polsos sinusoidals d'excitacio
26 -     xdc_excitation(Th, excitation);        % Establiment dels polsos que exciten el PMUT
27
28 -     distance = [0.5]*1e-3; %Meitat de la distancia entre transceptors
29 -     init_points = zeros(length(distance),2);
30 -     points = [init_points distance];
31
32 -     hp = calc_hhp(Th, Th, points); %Calcul del senyal rebut pel transductor
33
34 -     t = 1/fs:1/fs:length(hp)/fs;      %Establiment del vector temps per fer el plot
35
36 -     cont = cont + 1;
37 -     subplot(2,1,cont)
38 -     plot(t,hp)
39 -     title(['Eco establert a ',num2str(2*distance*10^3), 'mm amb f = ', num2str(f0/10^6),'MHz'])
40 -     xlabel('Temps [us]')
41 - end
42 - xdc_free(Th) %Alliberament de la memòria on es guarda el transductor

```

Figura 2.6: Codi generat per realitzar i visualitzar un estímul de dues longituds d'ona.

Amb FIELD II els transductors s'originen sempre centrats a les coordenades d'origen (0,0,0), i no hi ha possibilitat de moure'ls. Per aquest motiu, i donat que ambdòs transductors són

idèntics, s'emplea el mateix array de PMUTs com a emissor i receptor.

Per sortejar aquest impediment informàtic en primer lloc s'estimula l'emissor, i en segon lloc es calcula l'eco que es genera a una distància establerta "d", tenint en compte que en establir aquesta distància la separació entre els elements emissor i receptor equival a $2d$.

Però per poder realitzar la simulació s'han d'especificar determinats paràmetres per tal que FIELD II treballi dins del medi que desitjem, com poden ser la velocitat del so en líquid (c en m/s) o l'atenuació acústica (en dB/m)^{1,2}.

S'inicia establint els paràmetres de treball, i a continuació s'incorpora la comanda de FIELD II *field_info* que permet visualitzar tota la informació referent a les condicions de treball del programa.

```

Speed of sound:      1500.0000 m/s
Sampling frequency:  400.0000 MHz
Whether to use attenuation: 1
Frequency independent attenuation is:      1.0000 dB/m
-----
Speed of sound:      1500.0000 m/s
Sampling frequency:  600.0000 MHz
Whether to use attenuation: 1
Frequency independent attenuation is:      1.0000 dB/m

```

Figura 2.7: Condicions de treball de FIELD II per cada iteració freqüencial.

Com s'observa als paràmetres recollits a 2.7, el camp s'ha generat de la manera desitjada.

Referent a la totalitat del codi 2.6, el fet d'obtenir un funcionament correcte verifica també el funcionament dels PMUTs en condició de transceptors, veure figura 2.8.

Amb els cursors establerts a 2.9, es recalcula la freqüència obtinguda, per assegurar que no hi ha errors en la transmissió. Es situen els cursors en màxims adjacents de la funció, temps equivalent a una longitud d'ona. La inversa de la diferència d'aquests valors aporta directament la freqüència mesurada, que resulta ser de 3.9 i 5.8 MHz corroborant una correcta emissió i recepció.

Aquests senyals s'enviaran a LTspice per seguir amb el procés de filtratge.

¹Flavius Pop et al. "Enabling Real-Time Monitoring of Intrabody Networks through the Acoustic Discovery Architecture". A: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* PP (juny de 2020), pàg. 1-1. DOI: 10.1109/TUFFC.2020.3002973.

²G. Enrico Santagati i Tommaso Melodia. "Experimental Evaluation of Impulsive Ultrasonic Intra-Body Communications for Implantable Biomedical Devices". A: *IEEE Transactions on Mobile Computing* 16 (gen. de 2016), pàg. 1-1. DOI: 10.1109/TMC.2016.2561277.

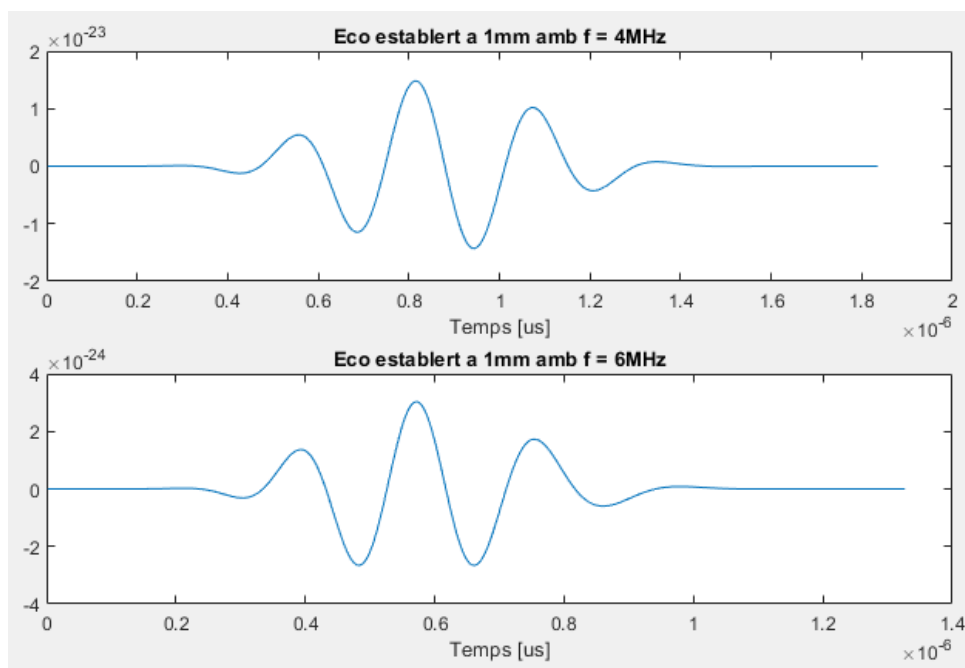


Figura 2.8: Gràfiques generades per les dues freqüències on es visualitzen els senyals captats pels transductors.

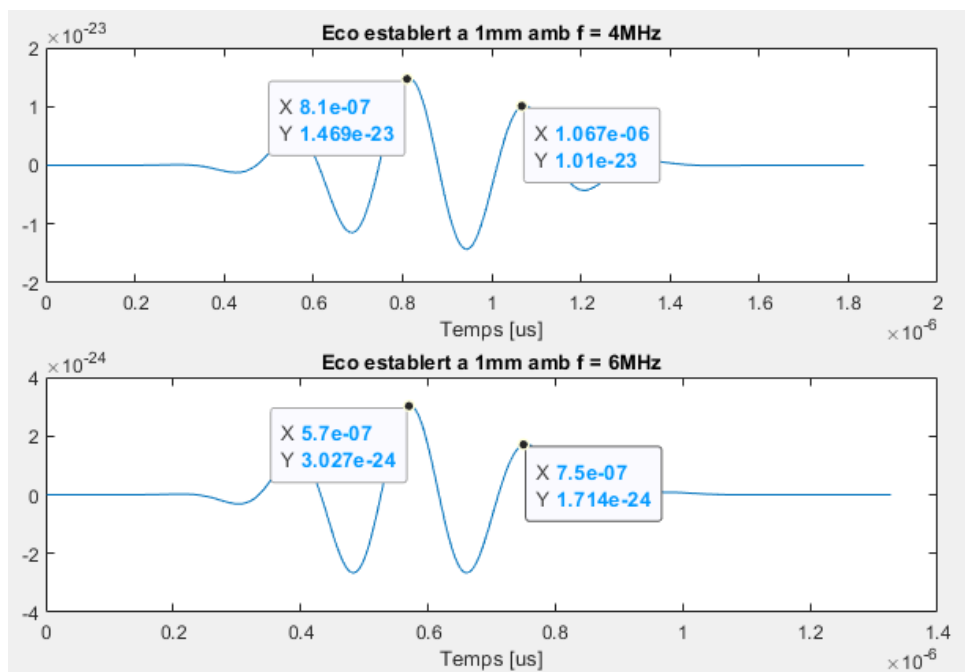


Figura 2.9: Figura 2.8 amb cursors per evaluar la freqüència obtinguda.

Capítol 3

Generació del filtre amb LTspice

Per generar els filtres en primer lloc s'han d'establir els requeriments que han d'assolir. Amb aquest objectiu es treballa amb freqüències centrals de 4 i 6 MHz, que permeten garantir una velocitat de transmissió suficient (fins a 4 i 6 Mbits/s teòrics, respectivament).

Amb intenció d'obtenir un sistema robust i evitar superposicions, s'estableixen rangs freqüencials amplis.

Per generar un sistema Full-Duplex es requereixen dos filtres passabanda. El primer permetrà el pas de freqüències entre 3.5 i 4.5 MHz, mentre el segon treballarà entre 5.5 i 6.5 MHz. D'aquesta manera hi ha 1 MHz de diferència entre el final del primer filtre i l'inici del segon.

Ambdós filtres tenen 1MHz de banda passant i seran generats amb polinomis Butterworth de 3r grau.^{1,2}

Per generar els filtres passa-banda es parteix del prototipus de filtre passa-baix que es veu a continuació.

Per aquest tipus de filtres s'acompleixen les fórmules següents:

$$g_k = 2 \sin \left(\frac{(2k-1)\pi}{2n} \right)$$

¹Electronicshub. *Butterworth Filter*. Febr. de 2019. URL: https://www.electronicshub.org/butterworth-filter/#%5C%EF%5C%BB%5C%BFButterworth_Approximations%5C%EF%5C%BB%5C%BF (cons. 22-08-2020).

²K.V.Puglia. *A General Design Procedure for Bandpass Filters Derived from Low Pass Prototype Elements: Part I*. des. de 2000. URL: <https://www.microwavejournal.com/articles/3102-a-general-design-procedure-for-bandpass-filters-derived-from-low-pass-prototype-elements-part-i#:~:text=A%5C%20low%5C%20pass%5C%20filter%5C%20may%5C%20be%5C%20converted%5C%20to,mapping%5C%20function%5C%20may%5C%20be%5C%20intuitively%5C%20or%5C%20mathematically%5C%20derived.> (cons. 22-08-2020).

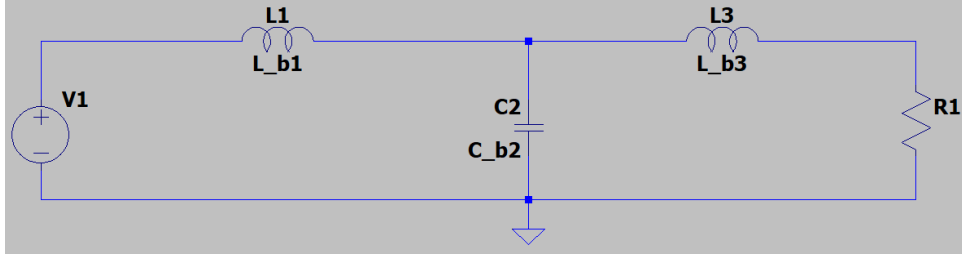


Figura 3.1: Estructura general d'un filtre de 3r ordre passa-baixos.

Donat que $n = 3$, s'obté directament

$$g_1 = g_3 = 1; \quad g_2 = 2$$

Per fer els filtres passa-banda s'inicia per una transformació dels components del filtre passa-baixos ideal 3.1, aquesta transformació es pot veure a continuació.

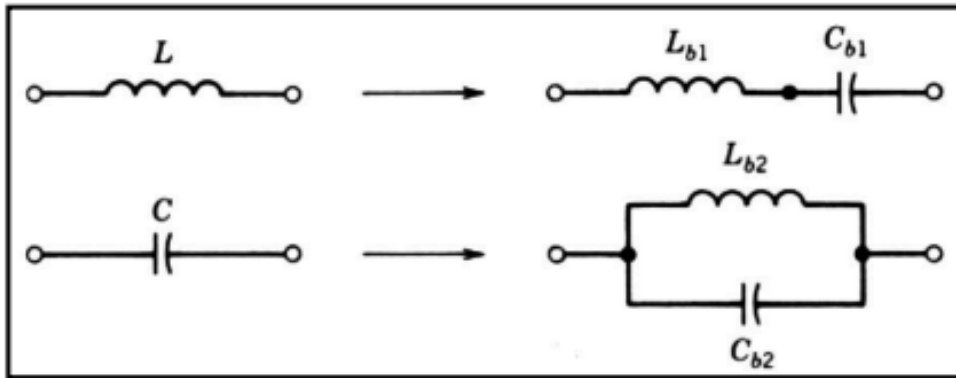


Figura 3.2: Transformació dels elements d'un filtre passa-baixos a un passa-banda.

Els valors dels elements de la figura 3.2 es calculen seguint les fòrmules següents:

$$L_{b1} = L_{b3} = \frac{g_1 R_0}{\omega_0 \Delta}; \quad C_{b1} = C_{b3} = \frac{\Delta}{g_1 R_0 \omega_0}; \quad L_{b2} = \frac{R_0 \Delta}{g_2 \omega_0}; \quad C_{b2} = \frac{g_2}{R_0 \omega_0 \Delta}$$

On ω_0 és la freqüència central i Δ és l'ample de banda fraccional.

Un cop es modifica el filtre 3.1 mitjançant la transformació de 3.2, s'obté un filtre passa-banda com el que es veu a continuació:

En el nostre cas, hem de generar dos filtres que hauran de ser units posteriorment.

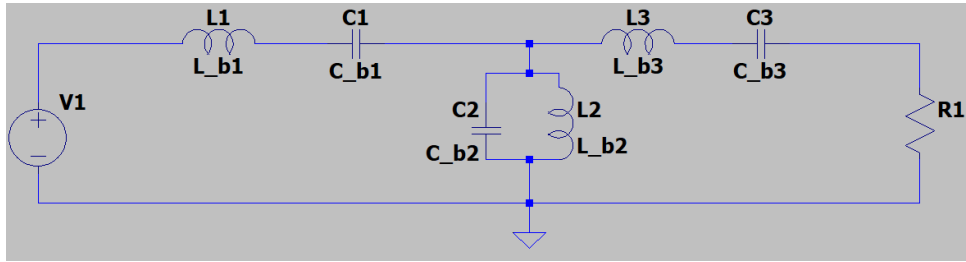


Figura 3.3: Filtre passa-banda general.

3.1 Primer filtre, a 4MHz

Per crear el primer filtre s'utilitzen les dades que sabem per calcular les capacitats i induccions necessàries. Aquest filtre tindrà una banda passant de 3.5 a 4.5 MHz. Això genera $\omega_0 = 4MHz$; $\Delta = \frac{1MHz}{4MHz} = 25\%$. Amb aquests valors i les fòrmules esmentades anteriorment s'obté la següent taula de capacitats i bobines:

$L_{b1} = L_{b3}$	$C_{b1} = C_{b3}$	L_{b2}	C_{b2}
$7.96\mu H$	$198.94pF$	$248.68nH$	$6.37nF$

Taula 3.1: Valors per generar el primer filtre passa-banda

Un cop s'obtenen les especificacions necessàries es pot generar el filtre amb LTspice.

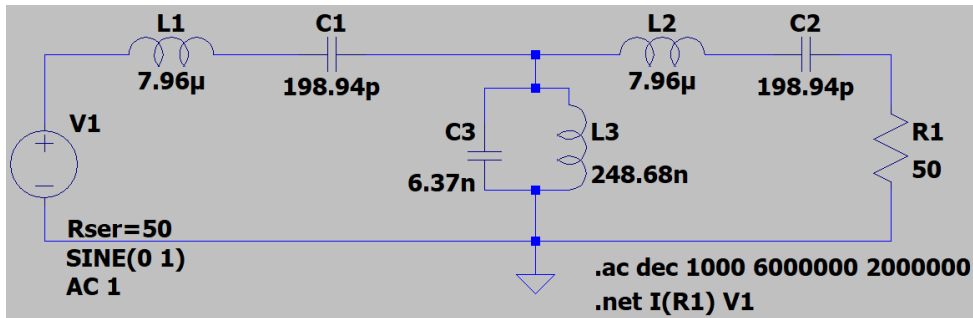


Figura 3.4: Filtre passa-banda de 3r ordre a 4MHz amb aproximació Butterworth.

En aquest moment s'ha generat el filtre desitjat. Si s'estudia el circuit 3.4 es pot observar que s'han establert resistències als ports d'entrada i sortida de 50Ω . Per corroborar que el filtre té el comportament desitjat es realitza una simulació entre 2 i 6 MHz amb la directiva ".ac". Per altra banda, per mesurar els paràmetres S s'emplea la directiva ".net" per establir els ports d'entrada i sortida del circuit.

A la simulació 3.5 es visualitzen els paràmetres S_{11} i S_{21} , que permeten analitzar el funcionament correcte del filtre, juntament amb uns cursors situats a -3dB, la freqüència de tall obtinguda.

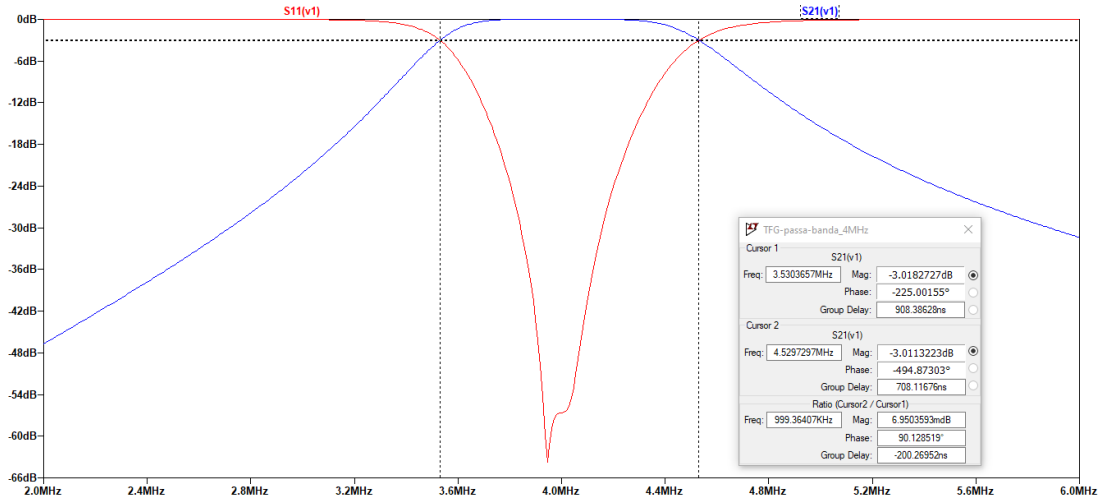


Figura 3.5: Paràmetres S_{11} i S_{21} del filtre 3.4.

Es comprova a 3.5, segons els cursors, que les freqüències de tall resultants es troben a 3.53 i 4.53 MHz.

Amb aquests resultats és possible iniciar la generació del segon filtre, a 6 MHz.

3.2 Segon filtre, a 6 MHz

Per aquest segon filtre es segueixen les pautes marcades al primer, però s'ha de tenir en compte que en aquest cas s'han modificat alguns valors. Per tant, s'han de recalculer les bobines i capacitats.

Aquest filtre tindrà una banda passant de 5.5 a 6.5 MHz. Això genera $\omega_0 = 6\text{MHz}$; $\Delta = \frac{1\text{MHz}}{6\text{MHz}} = 16.67\%$.

Un cop modificats els valors, es pot obtenir la taula pertinent.

$L_{b1} = L_{b3}$	$C_{b1} = C_{b3}$	L_{b2}	C_{b2}
7.96 μH	49.74pF	62.17nH	50.93nF

Taula 3.2: Valors per generar el segon filtre passa-banda

Amb aquests valors es genera el filtre desitjat.

En aquest cas (3.6) es segueixen generant els ports d'entrada i sortida igual, només cal modificar els components i les freqüències de simulació, entre 4 i 8 MHz.

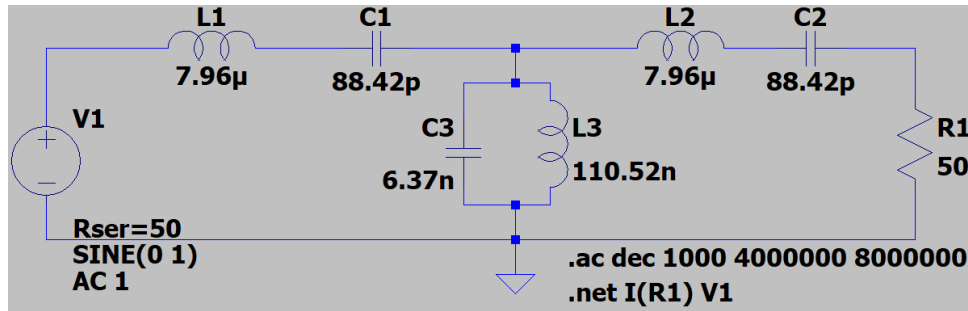


Figura 3.6: Filtre passa-banda de 3r ordre a 6MHz amb aproximació Butterworth.

Finalment, cal assegurar el correcte funcionament del filtre mitjançant la simulació pertinent.

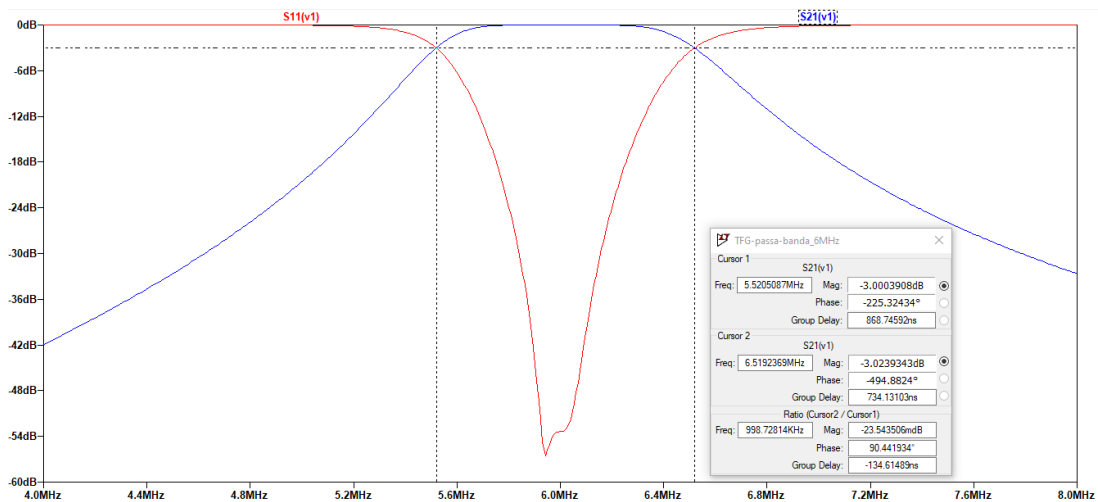


Figura 3.7: Paràmetres S_{11} i S_{21} del filtre 3.6.

De nou, es situen marcadors a -3dB. Es troben freqüències de tall a 5.52 i 6.52 MHz.

Donat que ambdós filtres presenten les característiques freqüencials desitjades i una banda de pas plana, es poden unir per generar el filtre que implementarà el comportament Full-Duplex del sistema.

3.3 Unió dels filtres generats

En aquest moment, cal unir correctament els filtres generats. Per això, s'inicia oblidant els ports d'entrada i sortida i col·locant els components dels dos filtres, un sota l'altre. Un cop s'han col·locat els elements, s'introdueix V1 com a entrada comuna, i es curtcircuiten les sortides dels filtres, donant lloc a la sortida del sistema.

La taula 3.3 conté la informació necessària per situar tots els elements on pertoca.

Filtre passa-banda	$L_{b1} = L_{b3}$	$C_{b1} = C_{b3}$	L_{b2}	C_{b2}
4 MHz	$7.96\mu\text{H}$	198.94pF	248.68nH	6.37nF
6 MHz	$7.96\mu\text{H}$	49.74pF	62.17nH	50.93nF

Taula 3.3: Valors per generar el filtre passa-banda complet.

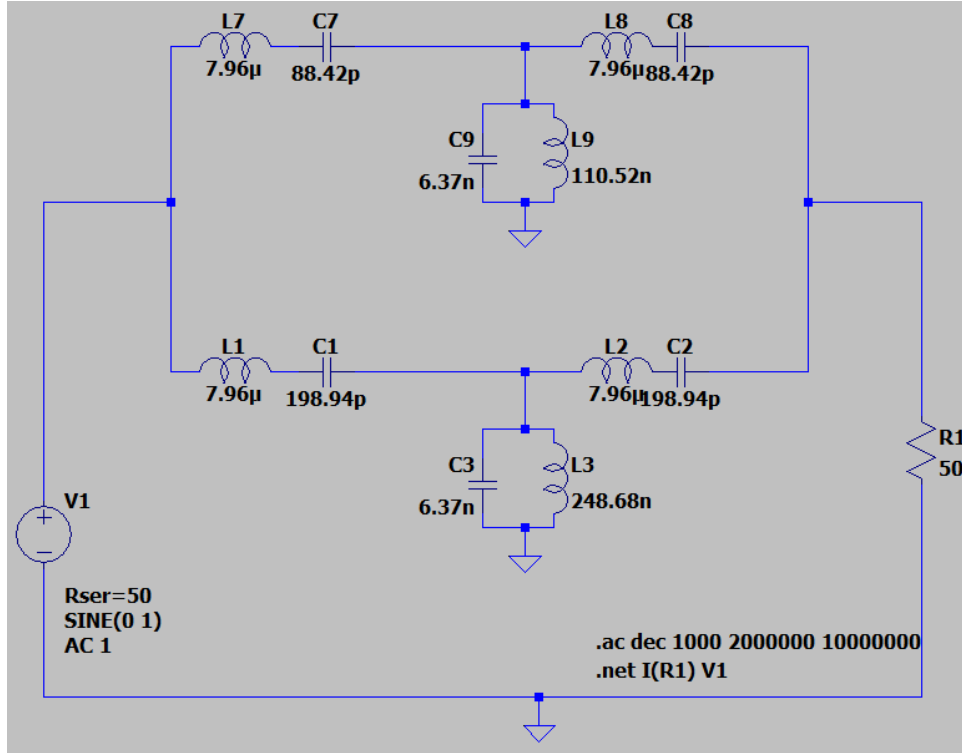


Figura 3.8: Filtre resultant de la combinació dels filtres 3.4 i 3.6.

Es poden observar les modificacions mencionades a la figura 3.8.

Un cop realitzat el muntatge del filtre, cal analitzar-ne el comportament. Per això, es simula el circuit de manera similar a les anteriors, entre les freqüències 2 i 8 MHz.

A continuació, es mostren dues simulacions (3.9 i 3.10) amb diferents cursors al primer i segon filtre, per comprobar si el seu comportament s'ha vist modificat per l'unió.

Es pot observar que s'imposen 4 freqüències de tall: 3.55, 4.6, 5.44 i 6.48 MHz.

Les freqüències no han patit cap canvi de relevància, però s'observa una banda de pas lleugerament menys plana. És necessari calcular quants dBs poden arribar a caure en la banda passant per saber si el filtre està actuant de manera correcta.

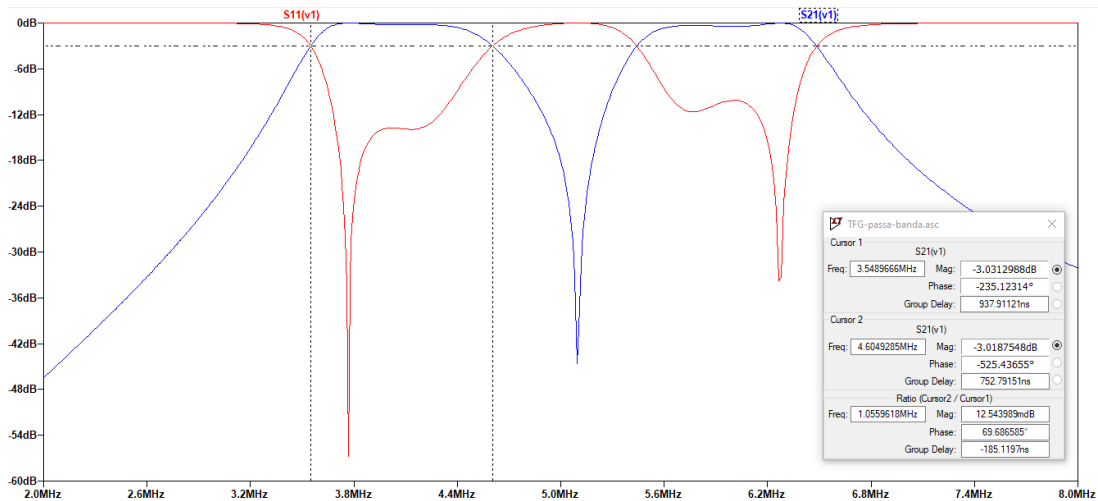


Figura 3.9: Simulació del filtre 3.8 amb cursors a les freqüències de tall del primer filtre.

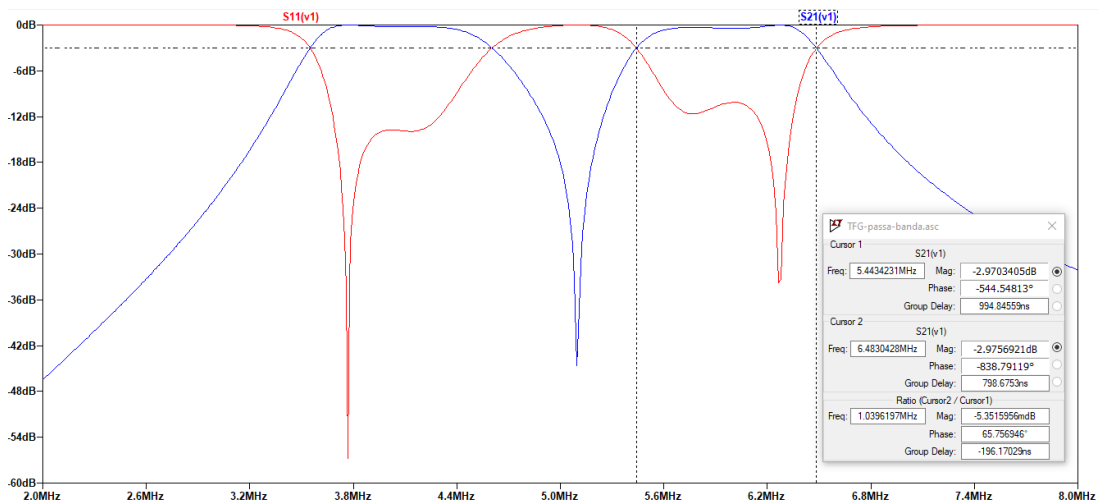


Figura 3.10: Simulació del filtre 3.8 amb cursors a les freqüències de tall del segon filtre.

Es realitza un augment a la banda de pas per comprobar-ne el comportament.

Un cop visualitzades les freqüències de màxima atenuació a la banda de pas a la figura 3.11, trobem que corresponen estretament amb la freqüències centrals. Es troben a 4.03 i 6.01 MHz. Tot i que la pèrdua més gran es produeix al segon rang freqüencial filtrat, s'obté un valor de -0.45dB en front dels -0.19dB del primer rang.

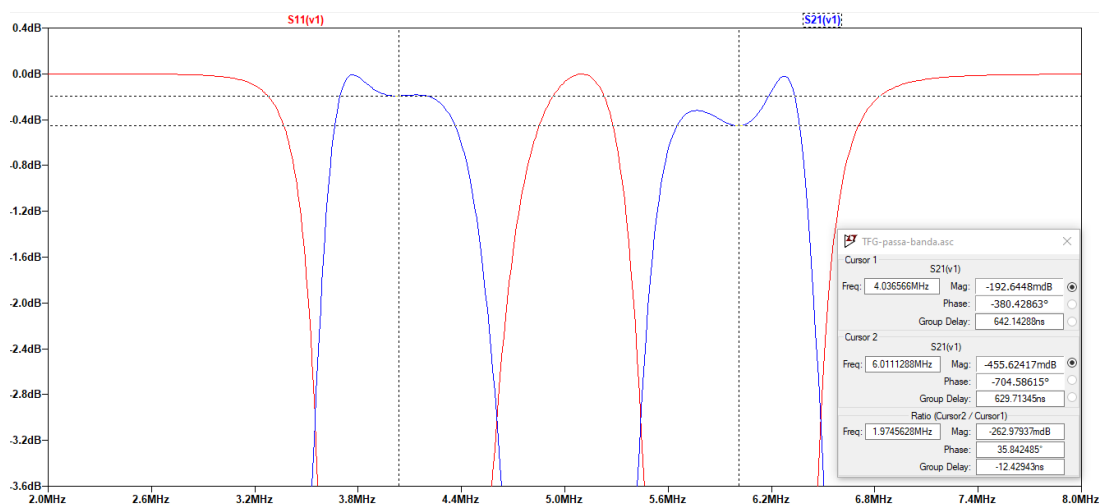


Figura 3.11: Augment a la banda passant de la simulació del filtre 3.8 amb cursors a les freqüències més atenuades.

Amb els resultats obtinguts, es pot concloure que hem generat el filtre correctament. Més endavant, es necessitarà aquest filtre per separar els senyals generats a Matlab.

Capítol 4

Enllaçament dels senyals generats a Matlab i LTspice

En aquest nou capítol es tractarà exclusivament la metodologia d'interacció entre els programes LTspice i Matlab (FIELD II inclòs). Per aquest motiu, els missatges enviats són dissenyats i coneguts, tenen la funció de corroborar que la transmissió es correcte en els dos sentits.

4.1 Esquema de funcionament de l'interfície LTspice-Matlab

Per poder realitzar les simulacions es realitzaran una sèrie de passos entre els programes Matlab i LTspice:

1. Generació dels missatges per les dues freqüències de treball amb Matlab.
2. Emissió i recepció dels missatges ultrasònics amb el programari FIELD II.
3. Transmissió de la informació continguda de Matlab a LTspice.
4. Generació de les ones de tensió i separació dels missatges mitjançant els filtres amb LTspice.
5. Retransmissió dels missatges filtrats de LTspice a Matlab.
6. Anàlisis del missatge resultant.

A la figura 3.11 es pot observar detalladament i gràfica el tractament de les dades realitzat per poder simular el sistema. Aquest flux correspon als passos enumerats prèviament.

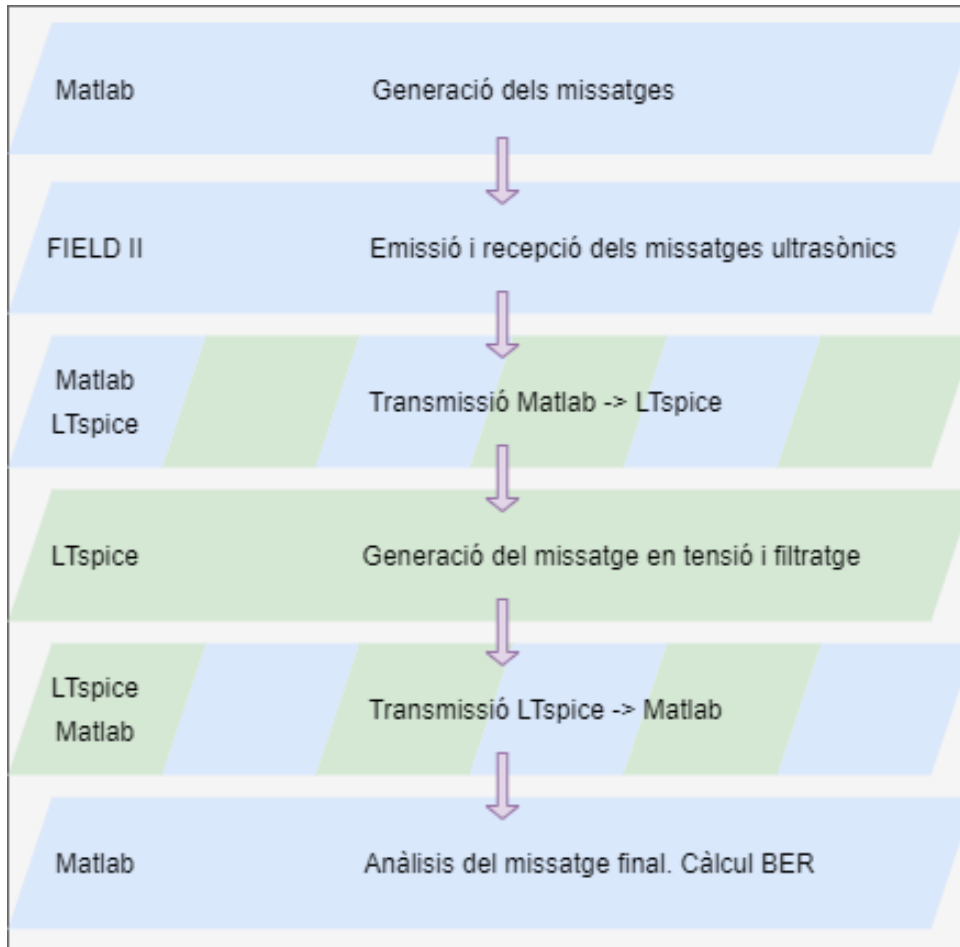


Figura 4.1: Esquema del tractament de dades general per simular el sistema de telecomunicacions.

4.2 Transmissió de dades de Matlab a LTspice

Per poder realitzar les transmissions d'informació de Matlab a LTspice s'ha de generar un arxiu d'extensió *.txt* que contingui l'ona obtinguda amb FIELD II. Per tal que LTspice pugui llegir correctament aquesta informació en un arxiu PWL (Piecewise Linear file), aquest haurà de tenir agrupades per parelles les dades de temps o tensió. És a dir, al *.txt* s'ha de llegir el següent ordre: $t_0 \quad V_0 \quad t_1 \quad V_1 \quad t_2 \quad V_2 \quad \dots \quad t_n \quad V_n$.

En aquest punt només s'estudiarà la generació d'un arxiu PWL que pugui ser llegit correctament per LTspice. Amb aquest objectiu es crea una funció denominada *crearPWL*, la qual demana els instants de temps i les corresponents tensions per generar un arxiu *.txt*.

És important esmentar que es treballa amb escales temporals i tensions molt petites. Com es pot observar al codi 4.2 (Línia 4), al *.txt* es generaran valors amb 12 dígits decimals en el temps, i 11 en la tensió.

```

function crear_PWL(data, t, filename)
    file_id = fopen(filename, 'w');
    for k = 1: length(data)
        fprintf(file_id, '%1.12f %2.11f ', t(k), data(k));
    end
    fclose(file_id);

```

Figura 4.2: Codi utilitzat per generar arxius *.txt* que LTspice utilitzarà per generar tensions per PWL.

Un cop creada la funció, es realitza un test amb dues senyals de diferent freqüència, però mateix comportament i nombre de punts.

```

clear all
close all

f1 = 4*1e6;
f2 = 6*1e6;
fs1 = 100*f1;
fs2 = 100*f2;
out1 = sin(2*pi*f1*(0:1/fs1:2/f1));
out2 = sin(2*pi*f2*(0:1/fs2:2/f2));

t1 = linspace(1,2/f1,2*(fs1/f1)+1);
t2 = linspace(1,2/f2,2*(fs2/f2)+1);

crear_PWL(out1,t1,'test1.txt');
crear_PWL(out2,t2,'test2.txt');

```

Figura 4.3: Prova de la funció *crearPWL*, genera els arxius *test1.txt* i *test2.txt*.

S'observa a 4.3 que es generen dues ones sinusoidals visibles a 4.4. Ambdues oscil·len sinusoidalment fins a 500 i 333 ns, respectivament. Aquesta diferència es deu a que es generen dues longituds d'ona en ambdós senyals, per tant la freqüència més alta finalitza en menys temps.

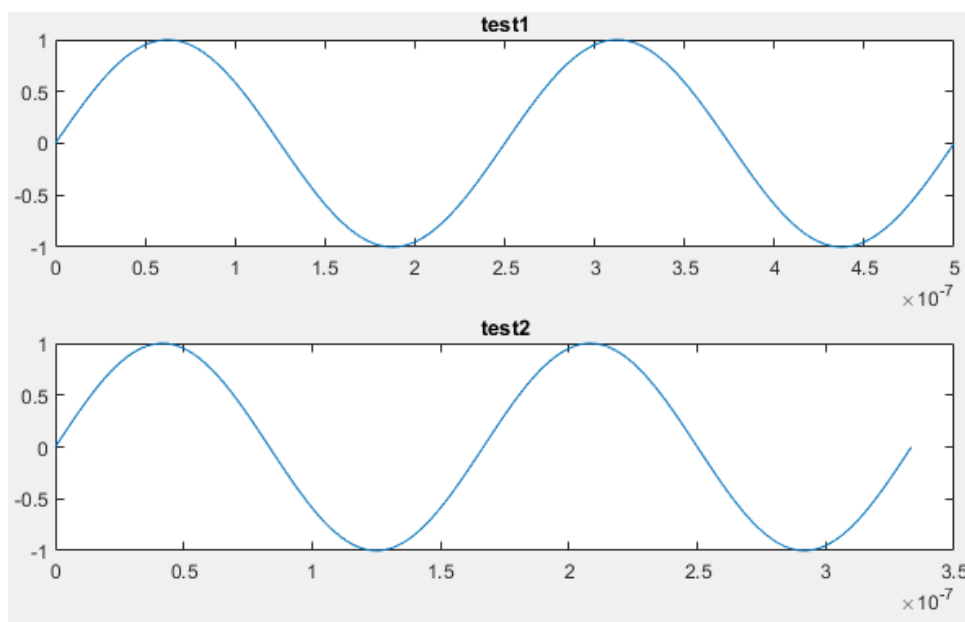


Figura 4.4: Plotting de les sortides generades per Matlab, *out1* i *out2*, per comparar amb les senyals captades a LTspice.

Per simular els senyals a LTspice es genera el següent esquemàtic molt senzill. Una font de tensió per cada arxiu PWL amb una resistència per generar caiguda de voltatge, així es pot simular la tensió de les fonts senzillament.

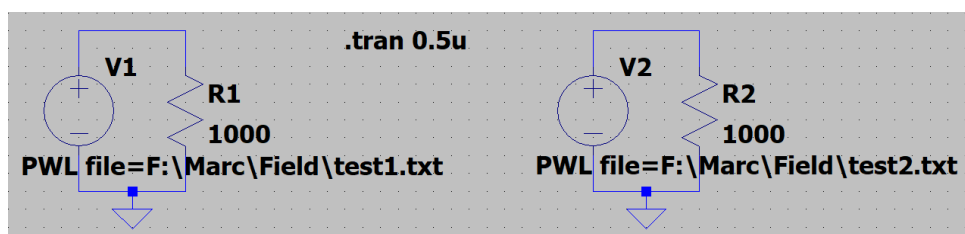


Figura 4.5: Esquemàtic per simular els arxius generats a Matlab.

S'inicia la simulació en transitori dels circuits i es visualitza a 4.6 que coincideixen plenament en forma i valor amb les de la figura 4.4. Ambdues tensions oscil·len sinusoidalment entre 1 i -1, i arriben a 500 i 333ns respectivament (mesurat amb cursor).

Per tant, s'ha comprovat que la funció *crearPWL* funciona de manera correcta i es té capacitat per transmetre informació en escales temporals petites a LTspice sense problemes.

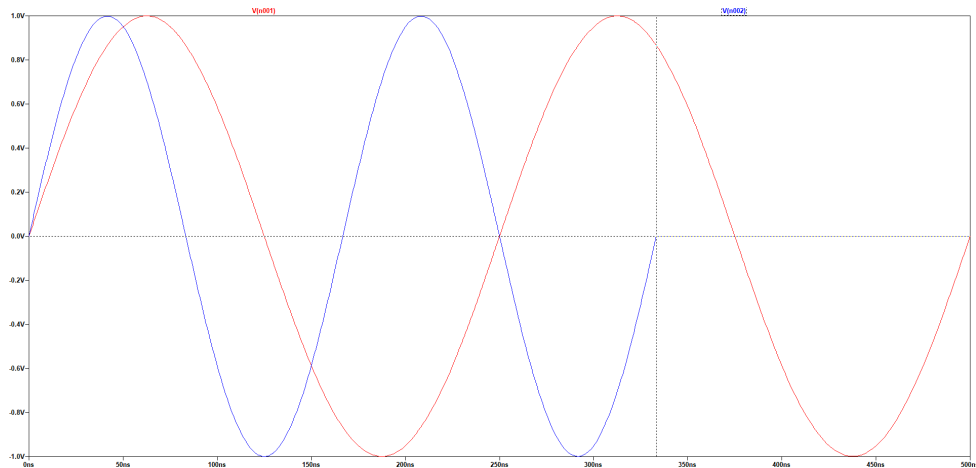


Figura 4.6: Simulació dels arxius PWL *test1*, en vermell, i *test2*, en blau.

4.3 Transmissió de dades de LTspice a Matlab

Llegir la informació de LTspice a l'entorn Matlab és senzill en comparació amb la resta de procediments efectuats. Això és gràcies a una funció disponible a MathWorks¹, esmentada *LTspice2Matlab*.

Per utilitzar-la, únicament s'ha d'assegurar que els arxius generats es guarden en la simulació al directori de Matlab on es treballa².

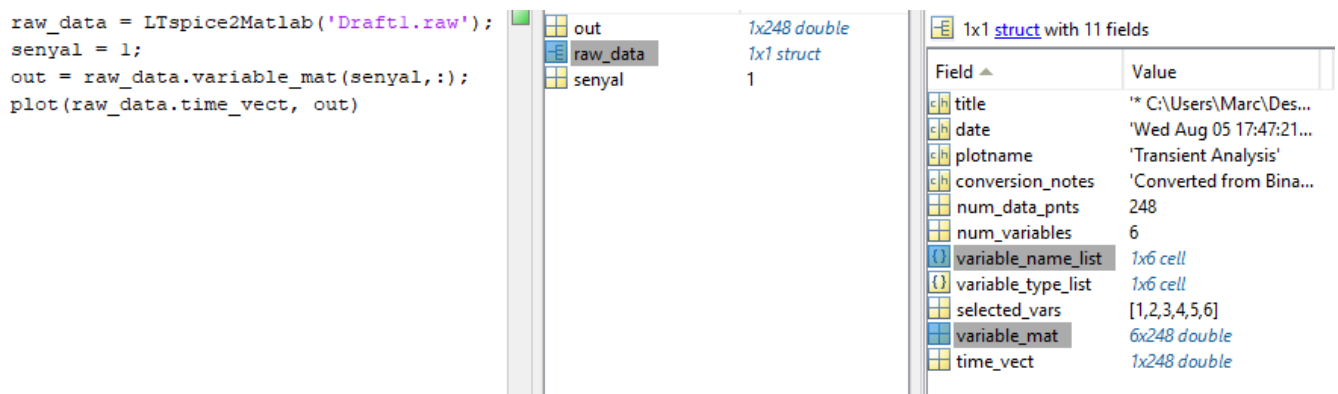


Figura 4.7: Dades generades amb l'ús de la funció *LTspice2Matlab*.

Es veu a la figura 4.7 com la funció esmentada genera una variable del tipus *struct* que conté tota

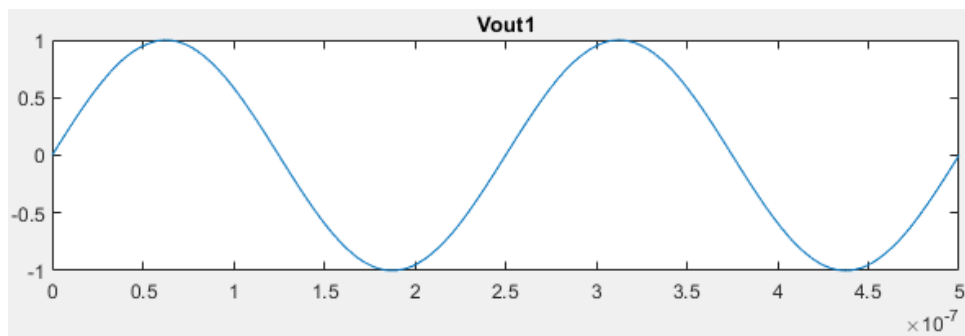
¹Paul Wagner. *Fast Import of Compressed Binary .RAW Files Created with LTspice Circuit Simulator*. Abr. de 2009. URL: <https://es.mathworks.com/matlabcentral/fileexchange/23394-fast-import-of-compressed-binary-raw-files-created-with-ltspice-circuit-simulator?focused=5113448&tab=function> (cons. 22-08-2020).

²Aaron Mattmiller. *Butterworth Filter*. Gen. de 2019. URL: <https://medium.com/@amattmiller/running-ltspice-from-matlab-630d551032cc> (cons. 22-08-2020).

raw_data.variable_name_list														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	V(vout1)	V(vout2)	I(R2)	I(R1)	I(V2)	I(V1)								
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														

raw_data.variable_mat														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0.0811	0.1229	0.1643	0.2054	0.2461	0.2864	0.3263	0.3854	0.4426	0.4977	0.5509	0.6020	0.6512
2	0	0.1214	0.1839	0.2452	0.3053	0.3641	0.4216	0.4780	0.5596	0.6350	0.7041	0.7669	0.8234	0.8736
3	0	1.2143e-04	1.8394e-04	2.4522e-04	3.0526e-04	3.6406e-04	4.2163e-04	4.7797e-04	5.5961e-04	6.3498e-04	7.0407e-04	7.6688e-04	8.2341e-04	8.736
4	0	8.1114e-05	1.2291e-04	1.6433e-04	2.0539e-04	2.4607e-04	2.8639e-04	3.2633e-04	3.8544e-04	4.4256e-04	4.9770e-04	5.5086e-04	6.0203e-04	6.512
5	0	-1.2143e-04	-1.8394e-04	-2.4522e-04	-3.0526e-04	-3.6406e-04	-4.2163e-04	-4.7797e-04	-5.5961e-04	-6.3498e-04	-7.0407e-04	-7.6688e-04	-8.2341e-04	-8.736
6	0	-8.1114e-05	-1.2291e-04	-1.6433e-04	-2.0539e-04	-2.4607e-04	-2.8639e-04	-3.2633e-04	-3.8544e-04	-4.4256e-04	-4.9770e-04	-5.5086e-04	-6.0203e-04	-6.512
7														
8														
9														
10														
11														
12														
13														
14														

raw_data.time_vect														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	3.2324e-09	4.8991e-09	6.5658e-09	8.2324e-09	9.8991e-09	1.1566e-08	1.3232e-08	1.5732e-08	1.8232e-08	2.0732e-08	2.3232e-08	2.5732e-08	2.823
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														

Figura 4.8: Contingut de les variables *variable_mat* i *variable_name_list* i *time_vect*.Figura 4.9: Gràfic de la tensió V_{out1} obtinguda de LTspice.

la informació de l'arxiu generat per LTspice. Concretament s'analitzen les matrius *variable_mat* i *variable_name_list*. La primera conté tota la informació de tots els atributs mesurables (voltatge, intensitat, paràmetres S...) mentres la segona conté la relació de cada fila de *variable_mat* amb el tipus de informació que conté (Vout, i(R1)...). A la figura 4.8 s'exemplifiquen de manera clara els continguts esmentats.

L'array *variable_name_list* conté els noms de cada variable, la matriu *variable_mat* conté les dades calculades de cada variable i *time_vect* conté els instants de temps corresponents.

Entesa l'estructura *raw_data* que s'obté amb la funció *LTspice2Matlab*, es genera al codi 4.7 V_{out1} , corresponent segons 4.5 a la tensió generada a 4.3 *out1* i plotejada amb Matlab a 4.4. Es realitza el plot de la nova tensió V_{out1} i es genera el gràfic 4.9, el qual és idèntic al gràfic *test1* de la figura 4.4, ja que són aquestes les dades originals que hem enviat i rebut entre Matlab i LTspice.

Per tant, amb els procediments citats durant aquest capítol s'han establert els mecanismes a utilitzar per transmetre la informació entre els diferents programes involucrats.

Capítol 5

Comprovacions del correcte funcionament del sistema de telecomunicacions.

En aquest capítol es realitzarà una simulació funcional del sistema de comunicació. Això vol dir que s'inclouran tots els apartats necessaris per la generació, transmissió, recepció i tractament d'ones. En aquest punt es pretén realitzar una simulació de valors coneguts per comprovar possibles errades en el procediment.

A l'últim capítol s'erradicaran les arbitrarietats en el missatge per tal d'obtenir simulacions més reals amb valors pseudo-aleatoris sotmesos a soroll gaussià.

5.1 Elaboració del missatge a emetre amb Matlab i FIELD II

En primer lloc s'introdueixen els bits a simular. Per tal d'estudiar el correcte funcionament del sistema de comunicacions és necessari visualitzar l'inici i el final de la transmissió, així com el canvi de valor de bit de 1 a 0 i viceversa.

Per aquests motius, el primer test del sistema es realitza enviant i rebent la cadena de bits anomenada *Bits_IN* (1, 0, 1).

És necessari analitzar les variacions del comportament freqüencial del senyal en passar pel filtre, donat que existeix la possibilitat que es necessitin un nombre mínim de longituds d'ona per obtenir un comportament adequat i evitar estats transitoris amb definició insuficient, tant al principi de l'emissió com a cada canvi de signe.

Per aquest motiu es modifica el senyal d'excitació visible a 2.6. Ara es genera en funció de

```

Bits_IN = [1,0,1];
excitation = [];
for i = Bits_IN
    if i == 1
        excitation = [excitation sin(2*pi*f0*(1/fs:1/fs:5/f0))];
    else
        excitation = [excitation -sin(2*pi*f0*(1/fs:1/fs:5/f0))];
    end
end

```

Figura 5.1: Codi utilitzat per generar els senyals a partir de la cadena de bits a enviar.

la cadena *Bits_IN*, seguint el codi de la figura 5.1. En aquest cas, s'ha augmentat el nombre de cicles del senyal a 5 per cada bit (anteriorment només 2) i es concatenen les excitacions corresponents a cada bit de la cadena. Així es simula el comportament transitori del conjunt FIELD II-LTspice a l'inici i amb cada canvi de 0 a 1 i 1 a 0.

Com s'ha mencionat al capítol 3, si cada bit es transmetés amb un sol cicle s'assolirien velocitats de 4 i 6 Mbits/s, en funció de la freqüència analitzada. Però, degut a que per cada bit s'utilitzen 5 cicles, aquestes velocitats es divideixen per aquest valor. S'assoleixen sota aquests paràmetres velocitats de transmissió de 800 i 1200 Kbits/s, per les freqüències de 4 i 6 MHz respectivament.

Aquesta acció modifica fortament les velocitats de transmissió citades al capítol 3. Es passa dels 4 i 6 Mbits/s teòrics a unes velocitats de 800 i 1200 Kbits/s, respectivament.

Tot i així, analitzant la velocitat mínima de transferència (800Kbits) menys un 25% de la velocitat en concepte de protocols, fluctuacions i reemissions de bits erronis, s'estableix un mínim de 600Kbits/s útils de transmissió, amb els que es podria arribar a transmetre informació, missatges de so i vídeo en resolució acceptable (640x480 a 30fps)¹. Aquesta velocitat inhereix transmissió de dades biomètriques o odomètriques en temps real.

A la figura 5.2 es poden observar les gràfiques enfrontades de l'excitació generada (15 longituds d'ona amb dos canvis de signe) i el senyal rebut un cop simulada la transmissió. Es confirma que el senyal generat segueix adequadament el patró inicial.

Si es comprova la tensió sempre al mateix interval de temps (cada longitud d'ona) s'observa que el resultat es màxim en els signes 1 i és mínim en els 0 degut al desfasament de 180° que imposarà el protocol BPSK. Per captar valors amb màxima robustesa és necessari esperar una longitud d'ona, tant a l'inici com als canvis de signe, ja que és el moment en que el senyal assoleix màxima amplitud.

En aquest punt és necessari realitzar alguns canvis per adaptar el codi a les noves condicions. Per poder separar al filtre els senyals de les dues freqüències primer és necessari sumar-los a

¹SocialGamers. *Tabla de Bitrates*. Gen. de 2017. URL: <https://www.socialgamers.es/tabla-bitrates-obs/> (cons. 21-08-2020).

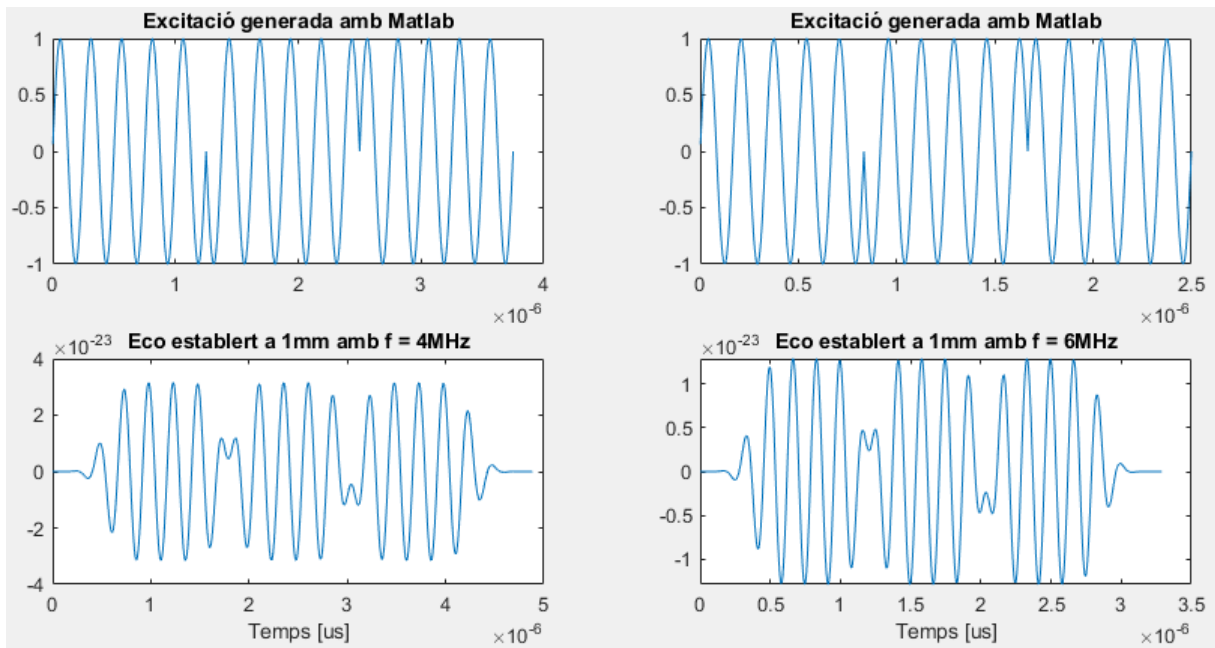


Figura 5.2: Gràfic de l'excitació generada enfront del senyal rebut pel PMUT.

Matlab.

Prèviament s'utilitzava un f_s diferent per cada freqüència central, pero no és possible sumar les tensions en temps discrets diferents. Per tant, s'unifica el vector temps mitjançant únicament la f_s més alta.

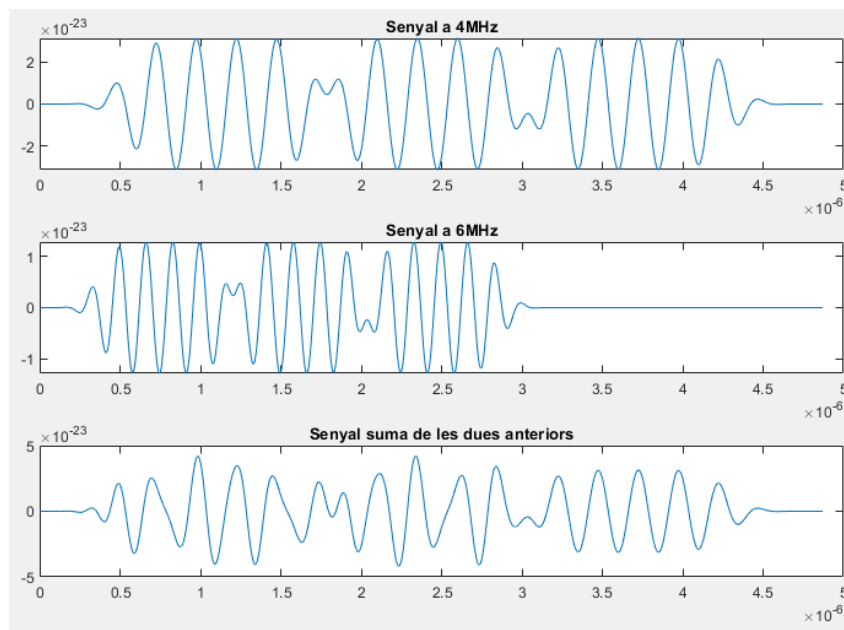


Figura 5.3: Senyals generats a 4, 6 MHz i la suma d'ambdós, respectivament.

Posteriorment, es sumen els senyals produïts i es crea l'arxiu *.txt* que generarà els valors de tensió en format PWL a LTspice.

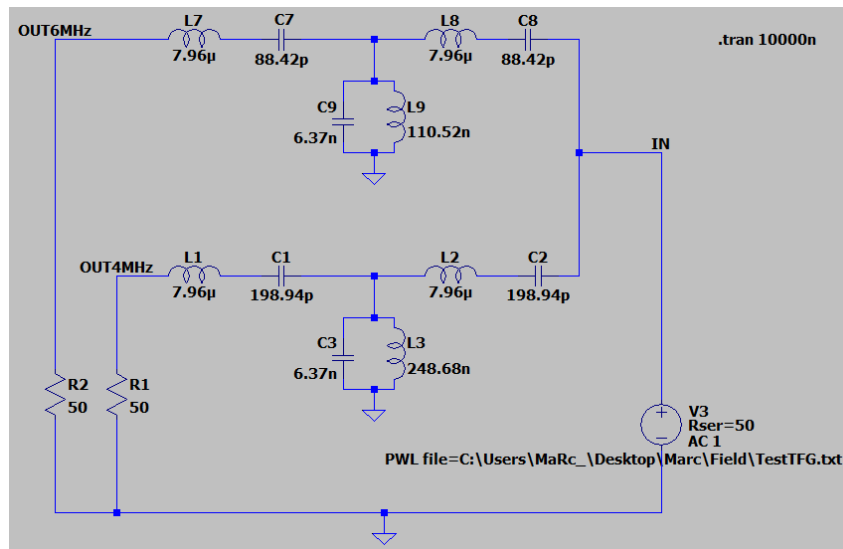


Figura 5.4: Filtre modificat per tenir com a entrada un senyal i sortides a les dues freqüències, 4 i 6 MHz.

Simultàniament, es modifica el filtre generat a 3.8. Per motius de disseny i per trobar els paràmetres S del mateix, s'establia la entrada a mà esquerra i la sortida a la dreta, i fent un anàlisis freqüencial en funció de V1 s'obtenien els paràmetres S.

Actualment es vol treballar en transitori, on insertem un senyal amb dues freqüències i es generen dues sortides, cadascuna amb una freqüència diferent. Per aquest motiu es realitzen les modificacions visibles a la figura 5.4, amb una entrada a mà dreta i una sortida per cada freqüència a l'esquerra. A més s'afegeixen noms als nodes importants, per la posterior visualització i obtenció de les dades amb Matlab.

Es llegeix amb LTspice l'arxiu PWL que conté la suma dels senyals i es realitzen les comprovacions pertinents, sent l'entrada igual a la tercera gràfica de la figura 5.3 i les sortides equivalents a les dues primeres.

Les figures 5.5, 5.6 i 5.7 corresponen respectivament a les gràfiques observades a 5.3. S'observa una gran similitud en les sortides a ambdues freqüències, que concorden perfectament en forma i magnitud.

Per assegurar que a les sortides hi ha un desfasament de 180° a cada canvi de signe es genera una ona pura a la freqüència específica (4 o 6 MHz), que servirà com a control. Sempre que l'ona filtrada coincideixi en fase amb el control es tractarà d'un 1, en cas d'un 0 s'ha de visualitzar un comportament negat respecte el control.

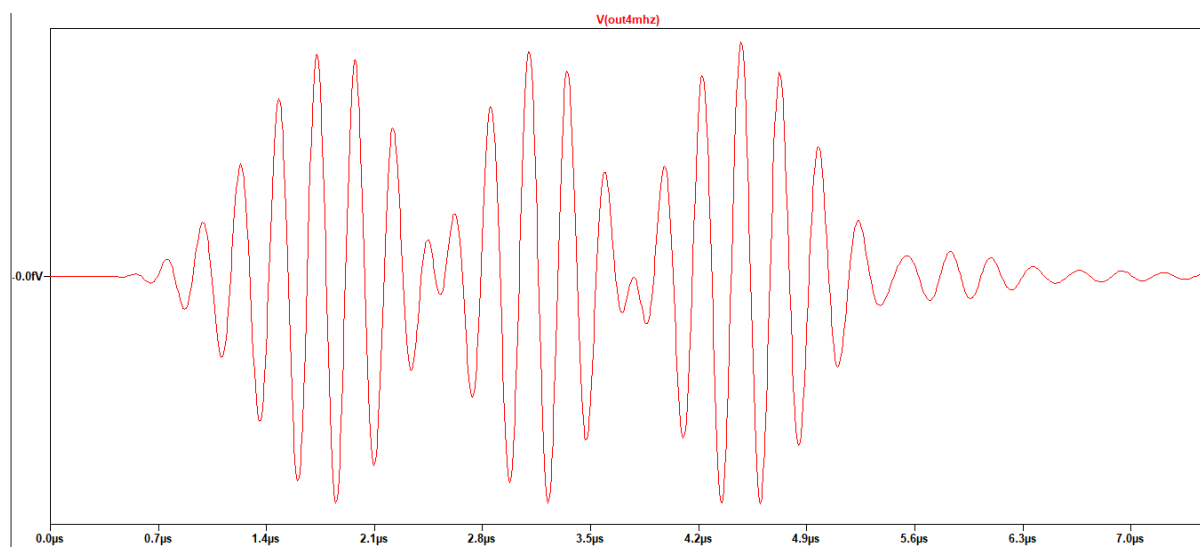


Figura 5.5: Sortida del filtre a 4MHz.

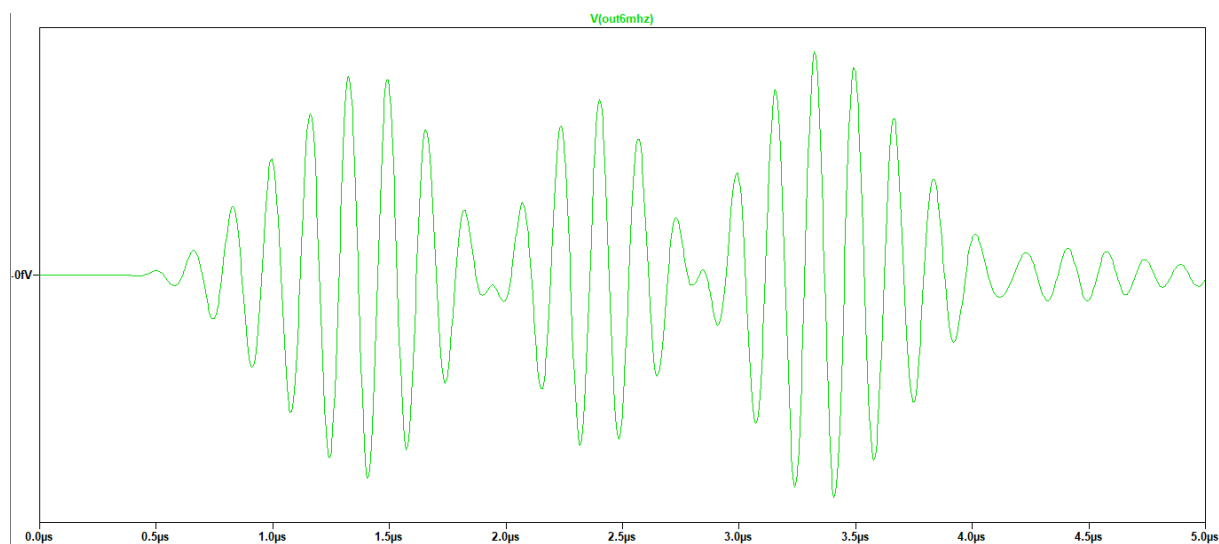


Figura 5.6: Sortida del filtre a 6MHz.

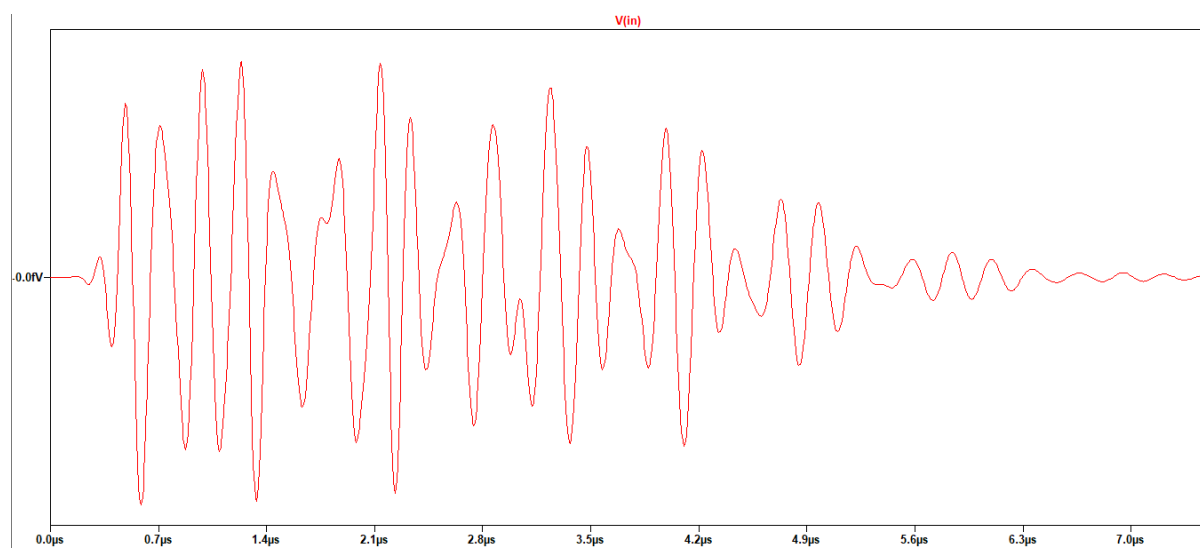


Figura 5.7: Entrada del filtre generada amb l'arxiu PWL.

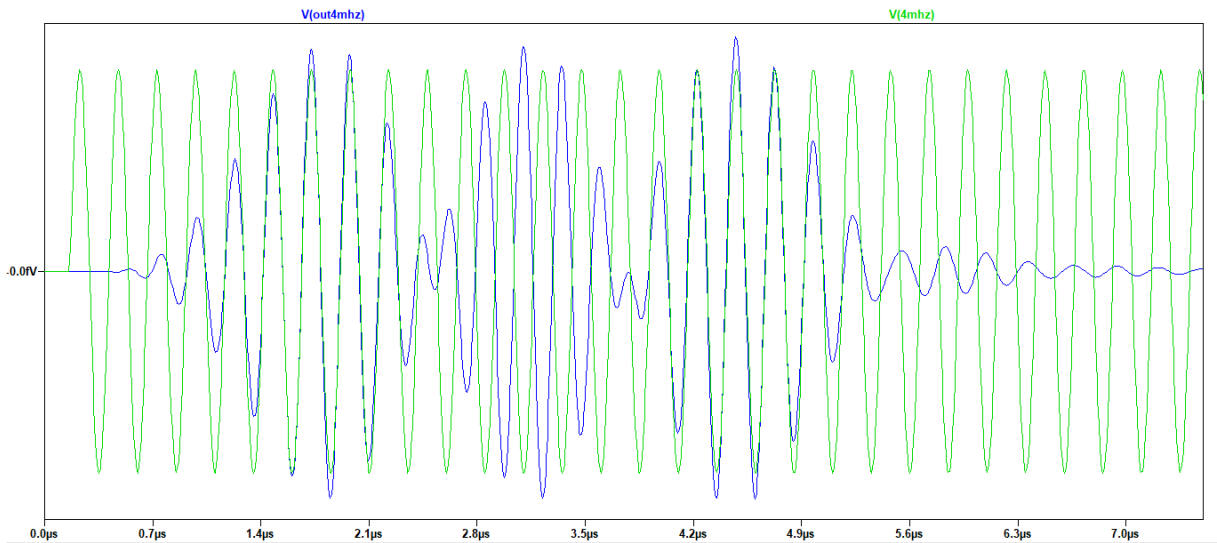


Figura 5.8: Sortida del filtre a 4MHz comparada amb un senyal sinusoidal pur a 4MHz.

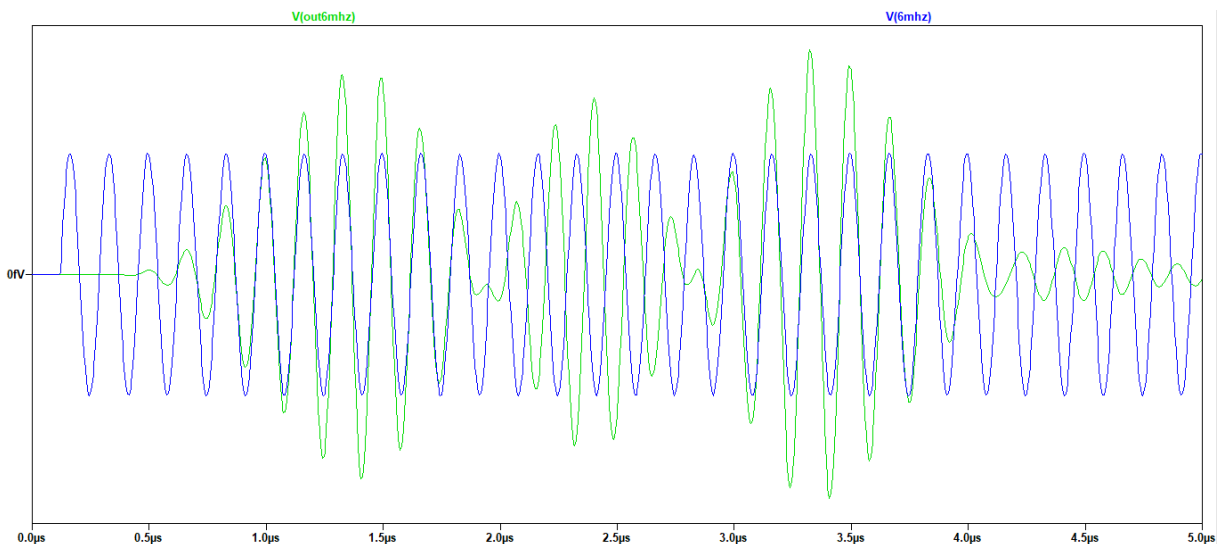


Figura 5.9: Sortida del filtre a 6MHz comparada amb un senyal sinusoidal pur a 6MHz.

Degut al tractament que realitza FIELD II sobre els senyals simulats, es genera un petit retard que cal conèixer per calcular el moment exacte en que s'han de mesurar els senyals per obtenir-ne els bits de sortida.

El retard esmentat té un valor, comprovat mitjançant les simulacions, de 165ns a 4MHz i 120ns a 6MHz. Donat que es tracta amb una ona periòdica, el retard real és major, exactament un nombre sencer de longituds d'ona més elevat.

Finalment es visualitza a les figures 5.8 i 5.9 que les sortides mantenen el canvi de bit generat per un desfasament de 180° .

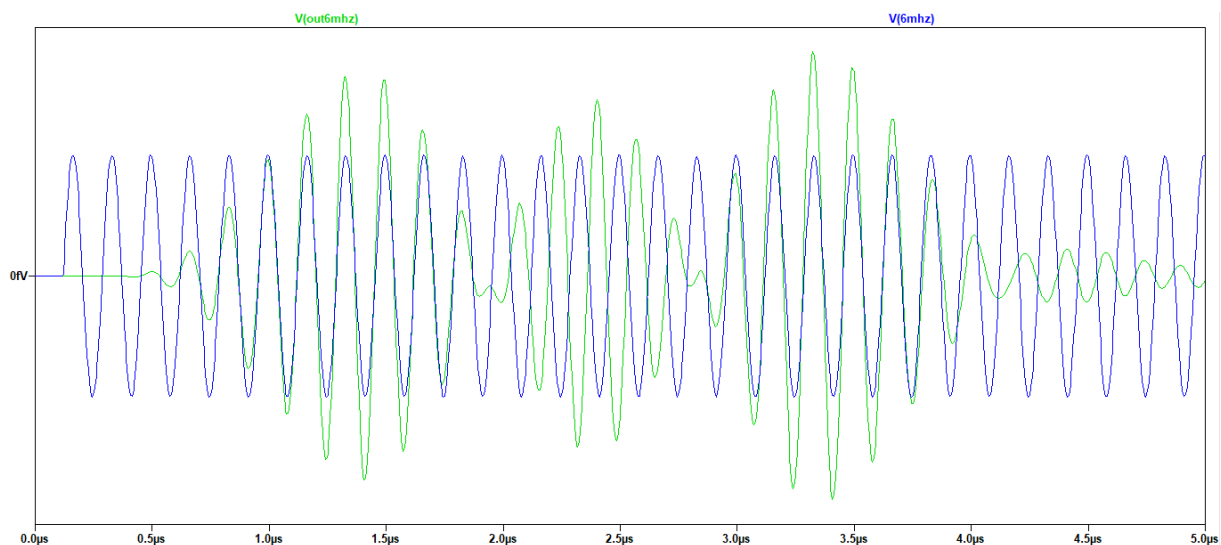


Figura 5.10: Dades provinents de la simulació en LTspice i tractades amb Matlab.

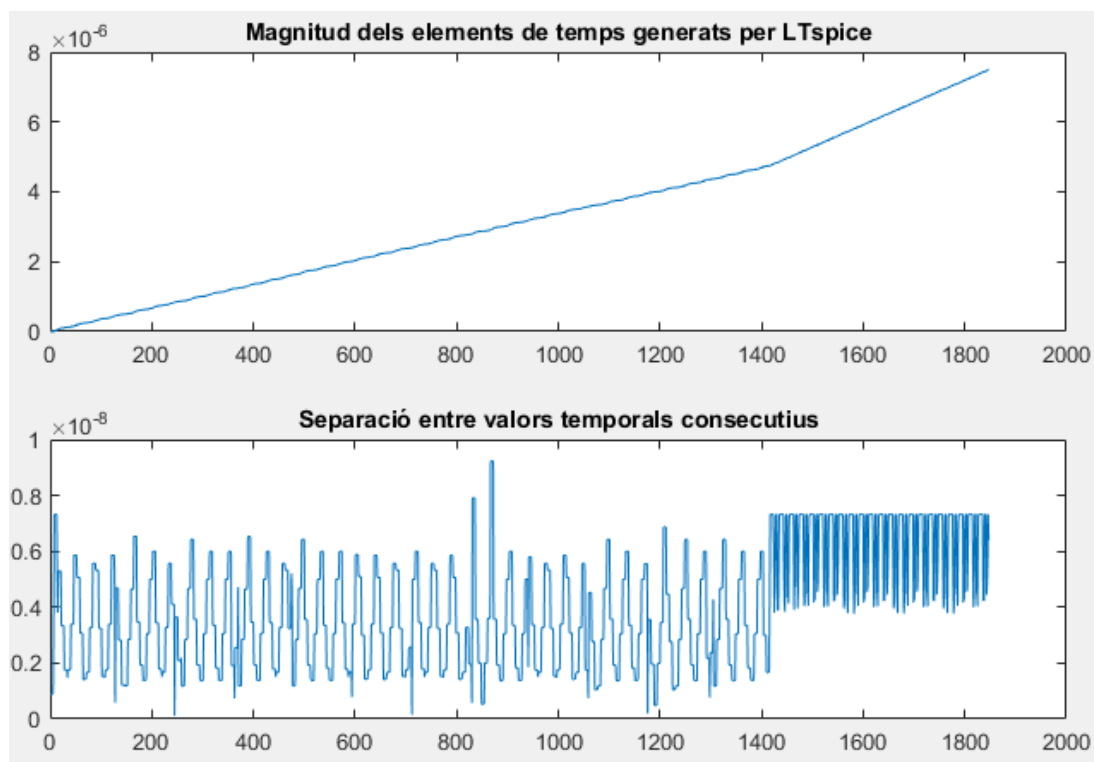


Figura 5.11: Demostració de la no-linealitat de les mostres temporals.

Per obtenir la cadena de bits inicial *Bits_IN* es tracta el senyal llegit de LTspice amb Matlab.

Desafortunadament, el tractament del senyal amb LTspice genera una resposta on els elements de l'array temps *raw_data.time_vect* no estan homogeniament espaiats. Com es veu als gràfics 5.11 (figura superior) els valors de temps no segueixen una tendència lineal. Per constatar que

no es pot determinar la manera de generació dels mateixos donat que les simulacions que realitza LTspice generen més o menys punts en funció de la derivada del senyal, de manera que estalvia cost computacional en trams on la funció no té grans canvis i augmenta l'esforç en trams d'alta variació.

A la gràfica inferior de 5.11 (figura inferior) es realitza el càlcul de l'espai que hi ha entre cada parella d'elements temporals. Si el temps fos espaiat de manera homogènia es visualitzaria una única línia recta.

Cal esmentar que aquest fenomen es degut al tractament intern de les FFT que calcula LTspice. És un procés intern en la base del programa que no és modificable, és a dir, no és possible generar una simulació temporal equiespaiada amb propòsits generals. Tot i així, és podria exigir un pas temporal suficientment petit que no necessités reducció en els trams més abruptes de la funció. En aquest últim cas el senyal obtingut estaria equiespaiat al pas mínim imposat, però el cost computacional escalaria ràpidament i es necessitarien temps de simulació molt més extensos.

En aquest moment és necessari establir un mecanisme de captació dels bits de sortida d'aquest senyal on les referències temporals no són suficientment estables ni lineals com per establir un criteri de temps en funció de la posició dels elements. És a dir, donades les particularitats del tractament temporal de LTspice, no es pot assumir que una longitud d'ona sempre ocupi 250 elements.

En primer lloc s'ha valorat realitzar una detecció d'envolupant, però no és factible degut a la impossibilitat de detectar un canvi de fase amb aquest mètode de manera fiable. Seria un bon mètode en un sistema amb modulació per amplitud del senyal.

Finalment, s'ha optat per un mecanisme de detecció de màxims del senyal i l'anàlisi de la fase del mateix. Aquest anàlisi es basa en imposar l'enviament a l'inici del missatge dels bits (1, 0) consecutius. Aquest establiment permet comparar la fase del primer bit conegut (1) amb la de tots els bits posteriors, sempre que estiguin en fase seran 1s, els bits en contrafase seran 0s. El 0 consecutiu al primer 1 es genera per no modificar el ratio entre 1s i 0s.

Donat que aquest codi no pot treballar de manera directa amb les posicions de l'array temporal, cal analitzar el sistema per trobar els valors desitjats. El codi complet per la freqüència de 4 MHz es pot visualitzar a la figura 5.12.

El mecanisme es basa essencialment en trobar els instants de temps on cada bit té amplitud màxima. Un cop trobats, es resta el temps del primer màxim, de manera que s'assegura que la posició "0" del nostre array actual es situa a un màxim de tensió d'un bit "1".

Per obtenir els bits, un cop realitzat el procediment mencionat al paràgraf anterior, és suficient amb obtenir la resta de la divisió dels temps màxims amb la longitud d'ona. Per això es codifica la divisió modular dels elements esmentats.

```

1 - bits_OUT = []; %Inicialització de l'array de sortida.
2 - t_offset = 1.094e-6; %Offset degut al retard del senyal.
3 - t_bit = 1250e-9; %Duració temporal de cada bit, 5 longituds d'ona.
4 - V_max= [];
5
6 - for i = 1:3 %Aquest for genera a t_max els valors temporals de màxima amplitud del senyal
7 -     t_bit_actual = find((t>t_offset + (i-1)*t_bit) & (t<t_offset + i*t_bit));
8 -     [max_value(i), index(i)] = max(outl(t_bit_actual));
9 -     t_max_index(i) = t_bit_actual(index(i));
10 -    t_max(i) = t(t_max_index(i));
11 - end
12
13 - t_max = t_max - t_max(1);
14 - modulus = mod(t_max, 0.250e-6);
15
16 - for i = 1:3
17 -     if (modulus(i) < 0.0625e-6) || (modulus(i) > 0.1875e-6)
18 -         bits_OUT = [bits_OUT 1];
19 -     else
20 -         bits_OUT = [bits_OUT 0];
21 -     end
22 - end
23
24 - bits_OUT

```

```

>> Deteccio_Bits

bits_OUT =

     1     0     1

fu >>

```

Figura 5.12: Codi generat per detectar els bits de sortida.

Donat que aquest és el cas concret del senyal a 4MHz es sap que la longitud d'ona és de 250ns, per tant la resta sempre ens donarà un valor entre 0 i 249. El valor 0 equival a un bit en fase, per tant els valors entre 187.5-250 ns i 0-62.5 ns es computen com un bit "1". la resta (entre 62.5 i 187.5 ns) indica que el valor per aquell temps es troba en contrafase, sent computat com un "0".

Finalment es mostra la sortida anomenada *Bits_OUT*, que coincideix exactament amb l'entrada *Bits_IN* generada anteriorment. Es tracta d'una cadena de bits (1, 0, 1).

D'aquesta manera s'ha demostrat que existeix la possibilitat de construir un senyal en funció dels bits que es vulguin transmetre, simular l'emissió, transmissió i recepció d'aquest senyal i obtenir els bits enviats a través del senyal rebut.

Capítol 6

Simulacions i anàlisis del sistema de telecomunicacions. Comprovació del BER en funció del SNR

L'últim capítol es centra en la generació de missatges pseudoaleatoris de 10000 bits i el càlcul del respectiu BER en diferents condicions.

6.1 Generació i captació d'un missatge aleatori sense soroll

L'esquema final per utilitzar el sistema segueix el següent patró:

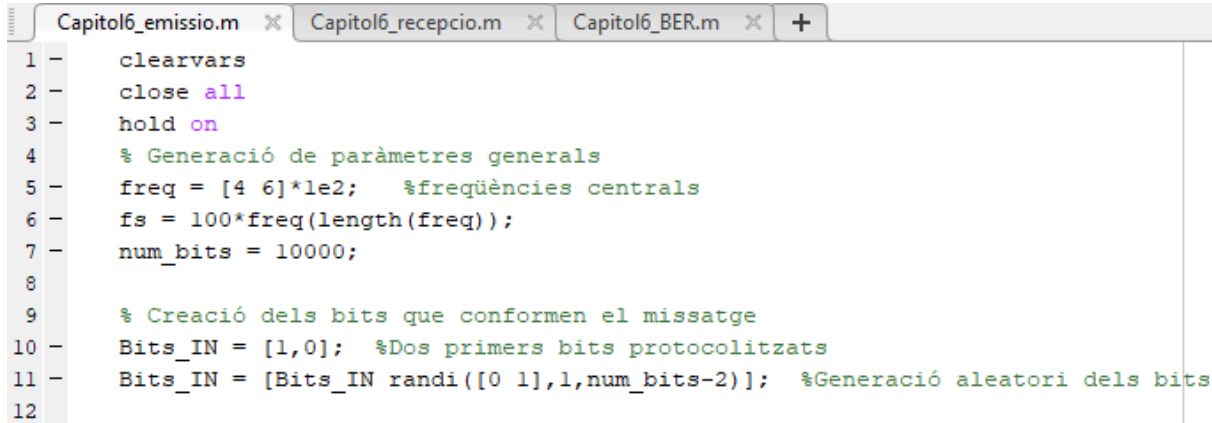
1. Generació dels bits d'entrada i del senyal a emetre. Simulació amb FIELD II.
2. Filtratge del senyal amb LTspice.
3. Captació de la informació de LTspice i obtenció dels bits de sortida.
4. Càlcul del BER per les dues freqüències.

Aquesta separació correspon a criteris d'ús. Per tal de fer servir el sistema cal que l'usuari intervingui un cop a cada etapa, clicant el botó d'inici en l'ordre adient.

Cal esmentar que aquest procés es podria automatitzar completament en un sol script de Matlab, utilitzant fitxers .bat per simular el filtre de LTspice des de Matlab. En aquest treball no s'ha considerat necessari complicar el codi i augmentar la possibilitat d'errors (en càlculs extensos el temps que es doni des de Matlab per tal que LTspice simuli pot ser molt variable) per guanyar en facilitat d'us a nivell usuari.

6.1.1 Generació dels bits d'entrada i del senyal a emetre. Simulació amb FIELD II

En els següents apartats es resumeix tot el codi utilitzat per realitzar les simulacions del sistema.



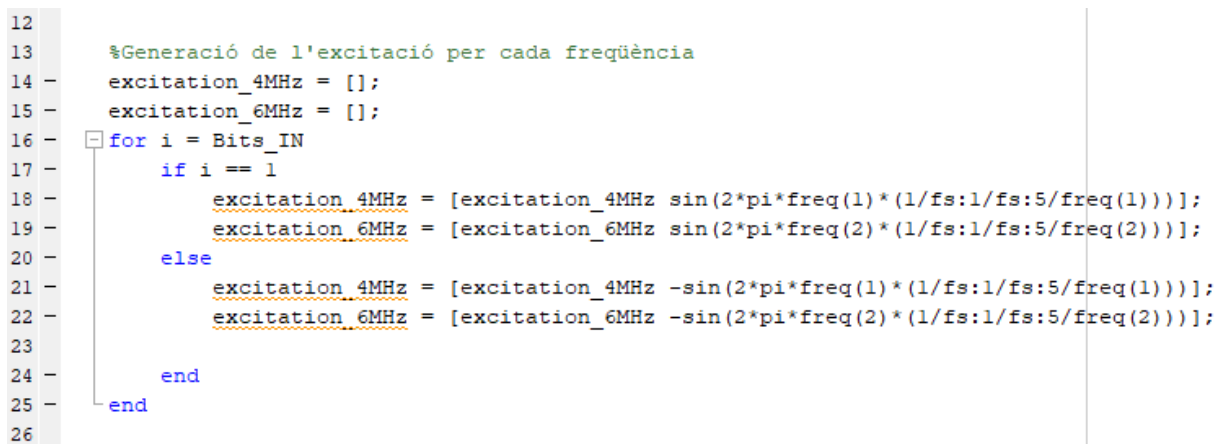
```

1 - clearvars
2 - close all
3 - hold on
4 - % Generació de paràmetres generals
5 - freq = [4 6]*1e2; %freqüències centrals
6 - fs = 100*freq(length(freq));
7 - num_bits = 10000;
8
9 - % Creació dels bits que conformen el missatge
10 - Bits_IN = [1,0]; %Dos primers bits protocolitzats
11 - Bits_IN = [Bits_IN randi([0 1],1,num_bits-2)]; %Generació aleatori dels bits
12

```

Figura 6.1: Establiment dels paràmetres generals i creació del missatge.

S'inicia inicialitzant els paràmetres generals i el missatge a transmetre. Com es veu a 6.1 s'especifica el nombre de bits que formaran la transmissió i es genera la cadena dels mateixos seguint el protocol d'inici de missatge (primers dos bits 1 i 0).



```

12
13 - %Generació de l'excitació per cada freqüència
14 - excitation_4MHz = [];
15 - excitation_6MHz = [];
16 - for i = Bits_IN
17 -     if i == 1
18 -         excitation_4MHz = [excitation_4MHz sin(2*pi*freq(1)*(1/fs:1/fs:5/freq(1)))];
19 -         excitation_6MHz = [excitation_6MHz sin(2*pi*freq(2)*(1/fs:1/fs:5/freq(2)))];
20 -     else
21 -         excitation_4MHz = [excitation_4MHz -sin(2*pi*freq(1)*(1/fs:1/fs:5/freq(1)))];
22 -         excitation_6MHz = [excitation_6MHz -sin(2*pi*freq(2)*(1/fs:1/fs:5/freq(2)))];
23 -     end
24 - end
25 - end
26

```

Figura 6.2: Generació de l'excitació del PMUT per cada una de les freqüències.

Es genera la funció d'excitació per cada freqüència en funció dels bits d'entrada, la cadena *Bits_IN*.

A continuació (6.3), es produeix l'excitació que s'aplicarà al PMUT sumant les dues calculades prèviament. Simultàniament es genera el vector temps que l'acompanya.

```

26
27 %Obtenció de l'excitació suma de les dues freqüències
28 - excitation_6MHz = [excitation_6MHz zeros(1,length(excitation_4MHz) - length(excitation_6MHz))];
29 - excitation = excitation_4MHz + excitation_6MHz;
30
31 %Generació del vector temps
32 - t = 1/fs:1/fs:length(excitation)/fs;
33

```

Figura 6.3: Vector d'excitació combinat per les dues freqüències.

```

33
34 %Establiment dels paràmetres del simulador FIELD II
35 - set_field('c', 1500); %velocitat del so [m/s]
36 - set_field('att', 1); %atenuació del medi
37 - set_field('use_att',1); %Imposem us d'atenuació
38
39 %Crida a la funció de generació del PMUT
40 - Th = Square_Transducer(); %apertura d'emissió i recepció
41

```

Figura 6.4: Establiment dels paràmetres de FIELD II i creació del PMUT.

S'estableixen els criteris relatius a la simulació en FIELD II. Aquests consten de la velocitat del so en el medi (1500m/s), l'atenuació de l'ona (1dB/mm)¹ i la creació del PMUT pròpiament, seguint la funció *Square_Transducer()* vista a la figura 2.2.

```

41
42 %Resposta impulsional
43 - impulse_response = sin(2*pi*((freq(1)+freq(2))/2)*(0:1/fs:2/((freq(1)+freq(2))/2)));
44 - impulse_response = impulse_response.*hanning(max(size(impulse_response))');
45 - xdc_impulse(Th,impulse_response)
46
47 %Establiment de l'excitació calculada pel PMUT
48 - xdc_excitation(Th, excitation);
49

```

Figura 6.5: Aplicació al PMUT de la resposta impulsional típica i l'excitació calculada.

Amb les commandes de FIELD II "xdc..." s'apliquen al transductor els paràmetres desitjats. En aquest cas es tracta d'una resposta impulsional típica i de l'excitació calculada prèviament a 6.3.

A 6.6 es crea el vector distància, cal recordar que el seu valor ha de ser de la meitat de la longitud desitjada degut a que es calcula l'eco del senyal emès. Simultàniament es calcula el senyal resultant de la simulació i es genera el vector temps que l'acompanya.

¹G. Enrico Santagati i Tommaso Melodia. "Experimental Evaluation of Impulsive Ultrasonic Intra-Body Communications for Implantable Biomedical Devices". A: *IEEE Transactions on Mobile Computing* 16 (gen. de 2016), pàg. 1-1. DOI: 10.1109/TMC.2016.2561277.

```

49
50     %Generació de la distància entre emissor i receptor
51 -    distance = 0.5*1e-3;    %Meitat de la distancia entre transceptors
52 -    init_points = zeros(length(distance),2);
53 -    points = [init_points distance];
54
55     %Simulació del senyal en funció de l'excitació i el PMUT
56 -    senyal = calc_hhp(Th, Th, points);
57
58     %Establiment del vector temps per enviar la simulació a LTspice
59 -    t_senyal = 1/fs:1/fs:length(senyal)/fs;
60

```

Figura 6.6: Establiment de la distància de comunicació i simulació del senyal resultant.

```

60
61     %Alliberament de la memoria on es guarda el transductor
62 -    xdc_free(Th)
63
64     %Creació de l'arxiu PWL per llegir amb LTspice
65 -    nom = strcat('Simulacio_TFG.txt');
66 -    crear_PWL(senyal, t_senyal, nom);

```

Figura 6.7: Alliberament de memòria i generació de l'arxiu PWL.

Amb la comanda `xdc_free` s'allibera la memòria que contenia la informació del transductor. És una línia necessària per evitar errors en compilacions consecutives. A més, es genera l'arxiu PWL esmentat *Simulacio_TFG.txt* mitjançant la funció generada a la figura 4.2, *Crear_PWL*.

6.1.2 Filtratge del senyal amb LTspice

En aquest punt ja s'ha executat el primer script de Matlab, per tant és necessari acudir a LTspice per simular el filtratge del senyal rebut i permetre una comunicació Full-Duplex.

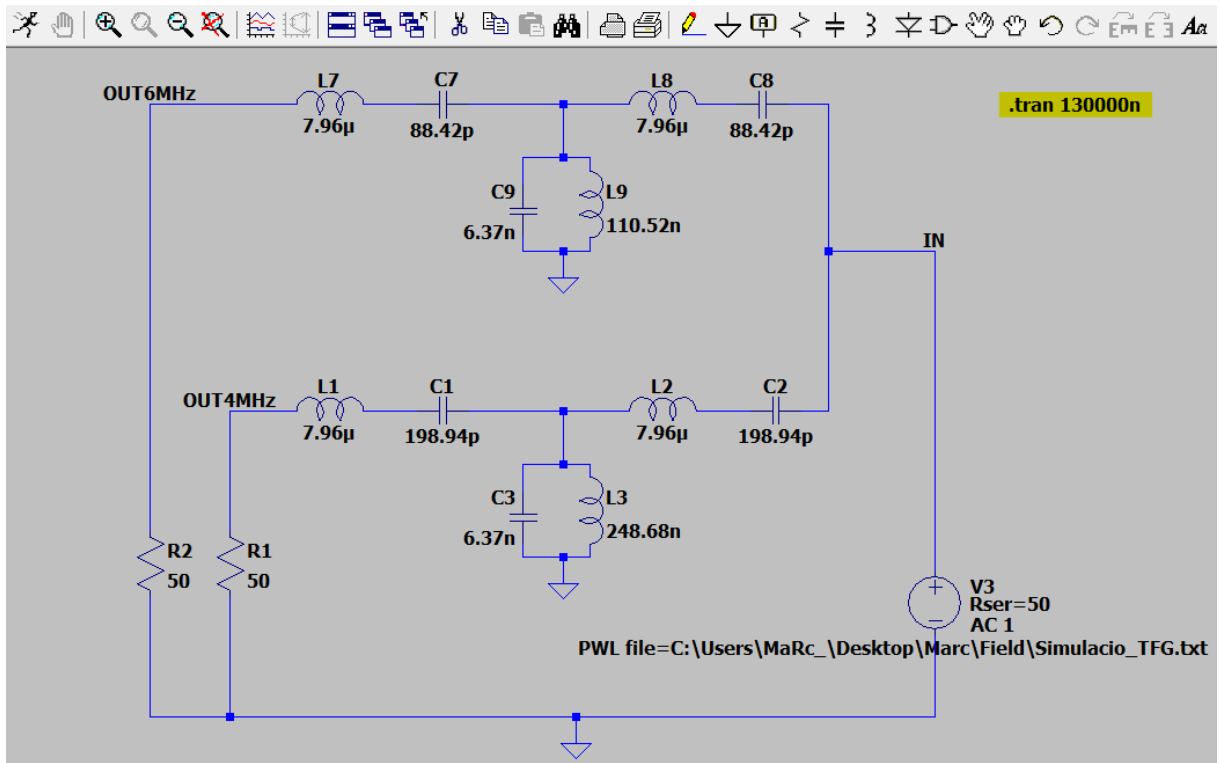


Figura 6.8: Filtre utilitzat a LTspice per separar les freqüències del senyal rebut.

Com es veu a la figura 6.8, el sistema és idèntic al visualitzat a 5.4, però es remarca en groc el valor que és possible que s'hagi de variar. Aquest valor correspon al temps final de simulació de LTspice, i depèn directament de la quantitat de bits que contingui la cadena *Bits_IN*.

6.1.3 Captació de la informació de LTspice i obtenció dels bits de sortida

En aquest segon script de Matlab es recull tota la informació necessària provinent de LTspice. Cal senyalar que es manté la variable *Bits_IN* per saber quin nombre de bits hem de llegir a la sortida, en un sistema de comunicació contínua no seria necessari.

Posteriorment es tracten els senyals de 4 i 6 MHz per extreure les cadenes de bits de sortida, *Bits_OUT_4MHz* i *Bits_OUT_6MHz*.

```

Capitol6_emissio.m  Capitol6_recepcio.m  Capitol6_BER.m  +
1 - clearvars -except Bits_IN
2 - close all
3
4 %Obtenció de les dades provinents de LTspice
5 - raw_data = LTspice2Matlab('test2TFG-passa-banda.raw');
6 - out1 = raw_data.variable_mat(5,:);
7 - out2 = raw_data.variable_mat(9,:);
8 - t_out = raw_data.time_vect;
9

```

Figura 6.9: Codi generat per extreure els senyals separats freqüencialment i el seu vector temps.

En aquest cas (6.9) s'utilitza el mateix codi que en 4.7, ampliat per captar els dos senyals en variables diferents. *out1* correspon a la variable 5 del *struct raw_data*, que conté el senyal a 4MHz. *out2* conté el senyal a 6MHz.

```

9
10 %Detecció dels bits a 4MHz
11 Bits_OUT_4MHz = []; %Inicialització de l'array de sortida.
12 t_offset = 1.094e-6; %Offset degut al retard del senyal.
13 t_bit = 1250e-9; %Duració temporal de cada bit, 5 longituds d'ona.
14 V_max = [];
15
16 for i = 1:length(Bits_IN) %Aquest for genera a t_max els valors temporals de màxima amplitud del senyal
17     t_bit_actual = find((t_out > t_offset + (i-1)*t_bit) & (t_out < t_offset + i*t_bit));
18     [max_value(i), index(i)] = max(out1(t_bit_actual));
19     t_max_index(i) = t_bit_actual(index(i));
20     t_max(i) = t_out(t_max_index(i));
21 end
22
23 t_max = t_max - t_max(1);
24 modulus = mod(t_max, 0.250e-6);
25
26 for i = 1:length(Bits_IN)
27     if (modulus(i) < 0.0625e-6) || (modulus(i) > 0.1875e-6)
28         Bits_OUT_4MHz = [Bits_OUT_4MHz 1];
29     else
30         Bits_OUT_4MHz = [Bits_OUT_4MHz 0];
31     end
32 end
33

```

Figura 6.10: Codi generat per detectar els bits de sortida del senyal a 4MHz.

Per solventar l'error de la manca de linealitat temporal, es dur a terme un procediment més complex.

En primer lloc, es calcula el *t_offset*, aquest implica el temps de retard que s'ha afegit al senyal inicial degut a les simulacions de FIELD II i LTspice. Per obtenir aquest valor s'ha buscat manualment el màxim de tensió del primer bit i se li han restat dues longituds ones i quart,

donat que el màxim es troba prop del punt mig i cada bit consta de 5 longituds d'ona. El valor d'una longitud d'ona correspon a l'inversa de la freqüència de 4MHz, 250ns.

La variable *t_bit* correspon al temps dedicat a 5 longituds d'ona. En aquest cas és de 1250ns.

El primer "for" busca, per cada bit, els índexos de temps que on es conté el bit en qüestió. Després, es busca el valor màxim en el vector *out1* dins de les posicions trobades al vector temporal. Finalment, es troba el valor temporal absolut (respecte l'inici del senyal temporal, no des del temps referit a un sol bit) i s'emmagatzema en un array fila. Quan acaba el "for" s'obté un vector fila on cada element conté la posició temporal del màxim de tensió de cada un dels bits.

A continuació es desplaça el vector temps de manera que comenci a contar en el màxim de tensió del primer bit. Sabent que es tracta d'un "1", es realitza la divisió modular de tots els elements de temps que indiquen els màxims de tensió entre la durada de la longitud d'ona. D'aquesta manera si són valors pròxims al 0 (pel funcionament d'una divisió modular, valors entre 0-62.5ns i 187.5-249ns) es tracta d'un 1 igual que l'utilitzat de control. En cas contrari es tracta d'un 0.

```

33
34     %Detecció dels bits a 6MHz
35 -   Bits_OUT_6MHz = []; %Inicialització de l'array de sortida.
36 -   t_offset = 0.9453e-6; %Offset degut al retard del senyal.
37 -   t_bit = 0.833333333333e-6; %Duració temporal de cada bit, 5 longituds d'ona.
38 -   V_max = [];
39
40 -   for i = 1:length(Bits_IN) %Aquest for genera a t_max els valors temporals de màxima amplitud del senyal
41 -       t_bit_actual = find((t_out > t_offset + (i-1)*t_bit) & (t_out < t_offset + i*t_bit));
42 -       [max_value(i), index(i)] = max(out2(t_bit_actual));
43 -       t_max_index(i) = t_bit_actual(index(i));
44 -       t_max(i) = t_out(t_max_index(i));
45 -   end
46
47 -   t_max = t_max - t_max(1);
48 -   modulus = mod(t_max, 0.166666666666e-6);
49
50 -   for i = 1:length(Bits_IN)
51 -       if (modulus(i) < 0.04166666666e-6) || (modulus(i) > 0.125e-6)
52 -           Bits_OUT_6MHz = [Bits_OUT_6MHz 1];
53 -       else
54 -           Bits_OUT_6MHz = [Bits_OUT_6MHz 0];
55 -       end
56 -   end

```

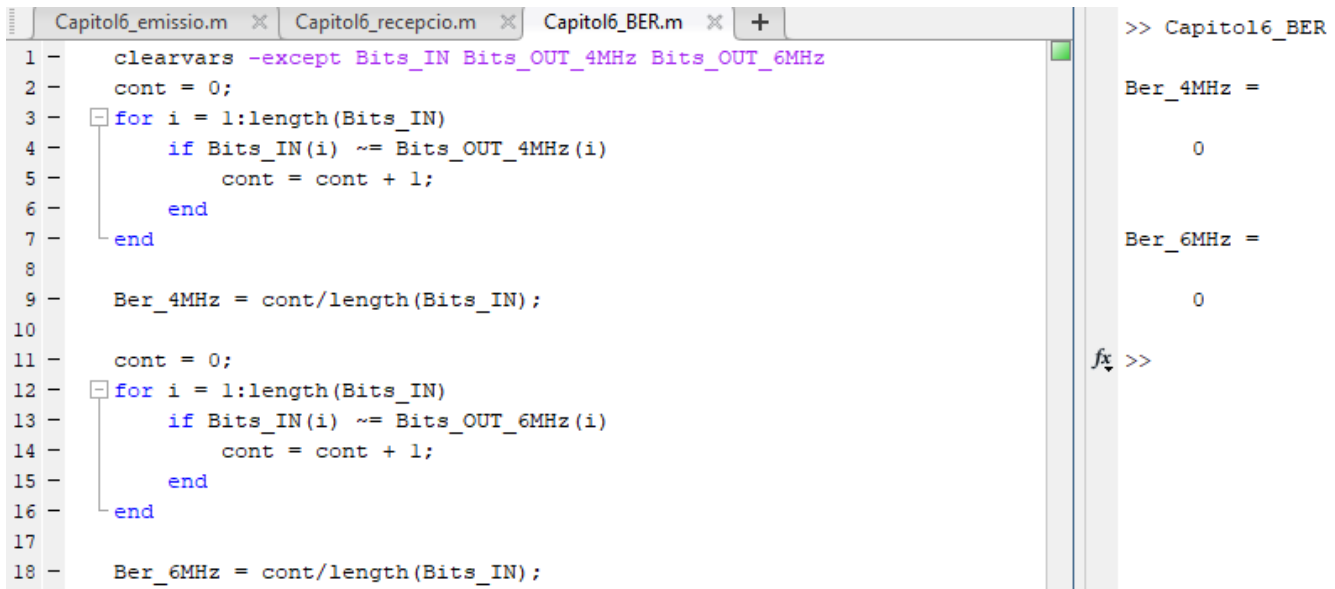
Figura 6.11: Codi generat per detectar els bits de sortida del senyal a 6MHz.

El funcionament en aquest cas és idèntic a l'explicat anteriorment. Únicament cal actualitzar les variables *t_offset* i *t_bit* per tal que treballin a 6MHz.

6.1.4 Càlcul del BER per les dues freqüències

Finalment, és necessari obtenir el BER per cadascuna de les freqüències treballades. D'aquesta manera es pot saber quina és la possibilitat de fallida de les comunicacions.

Donat que en aquest primer cas no s'ha aplicat cap tipus de soroll, en un sistema de comunicacions de funcionament òptim el BER deuria ser igual a 0.



```

Capitol6_emissio.m  Capitol6_recepcio.m  Capitol6_BER.m  +
1 - clearvars -except Bits_IN Bits_OUT_4MHz Bits_OUT_6MHz
2 - cont = 0;
3 - for i = 1:length(Bits_IN)
4 -     if Bits_IN(i) ~= Bits_OUT_4MHz(i)
5 -         cont = cont + 1;
6 -     end
7 - end
8 -
9 - Ber_4MHz = cont/length(Bits_IN);
10 -
11 - cont = 0;
12 - for i = 1:length(Bits_IN)
13 -     if Bits_IN(i) ~= Bits_OUT_6MHz(i)
14 -         cont = cont + 1;
15 -     end
16 - end
17 -
18 - Ber_6MHz = cont/length(Bits_IN);

>> Capitol6_BER

Ber_4MHz =

    0

Ber_6MHz =

    0

fx >>
  
```

Figura 6.12: Comprovació del BER per cada freqüència treballada.

En aquest cas s'ha treballat amb una cadena de 10000 bits. Com es pot observar el BER resultant és de 0 en els dos casos.

Queda demostrat que la codificació i el sistema funcionen de manera correcta en un entorn sense cap tipus de soroll.

6.2 Modificació del BER obtingut en funció del valor de SNR afegit al sistema.

Per afegir soroll al sistema es modifiquen les excitacions inicials abans de la primera simulació, de manera que el soroll es propagui a través de tot el sistema.

Per tal efecte s'incorporen les línies visibles a 6.13. Donat que les excitacions oscil·len entre -1 i 1, es genera el senyal de soroll que oscila entre $(-1 \text{ i } 1)/\text{SNR}$. A major SNR, menor és el soroll que es genera.

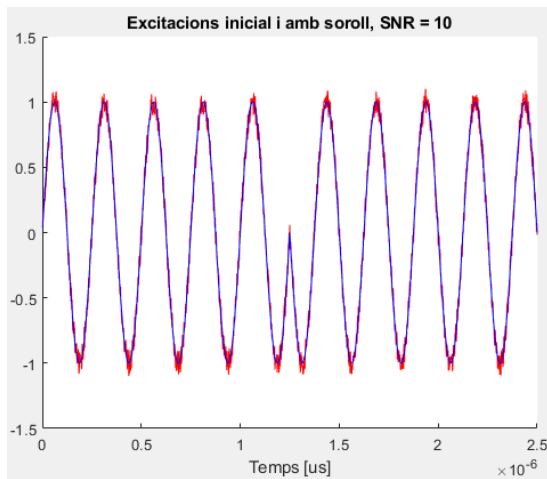
```
35 -      t = 1/fs:1/fs:length(excitation)/fs;
36
37      %Afegiment del soroll al sistema
38 -      SNR = 10;
39 -      excitation = awgn(excitation,SNR);
40
41      %Establiment dels paràmetres del simulador FIELD II
```

Figura 6.13: Línies utilitzades per incorporar el valor de SNR desitjat al sistema.

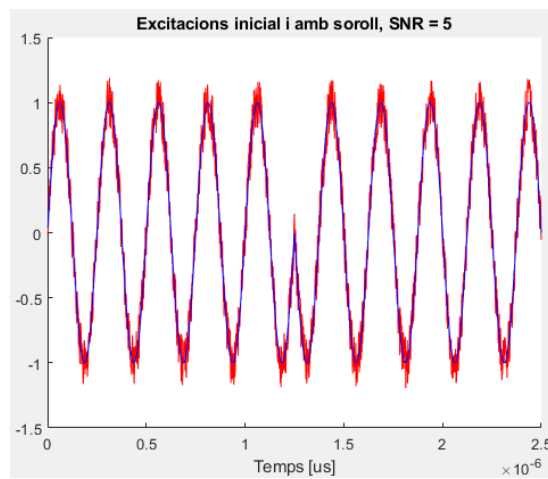
Per corroborar que el soroll es genera correctament es realitza un gràfic de l'excitació abans i després del soroll, per tots els valors de SNR estimats i per les dues freqüències.

Les dues pàgines a continuació mostren la variació de l'excitació del PMUT amb diferents valors de SNR. Aquests valors seran (en dB) 10, 7, 4.8, 3, 0 i -3.

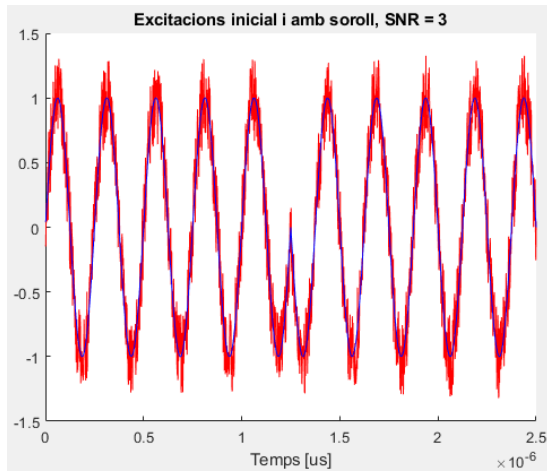
No es requereixen SNR menors, donat que el BER que s'introdueix en l'últim cas ja demostra que resulta inviable establir un sistema de comunicacions en aquestes condicions.



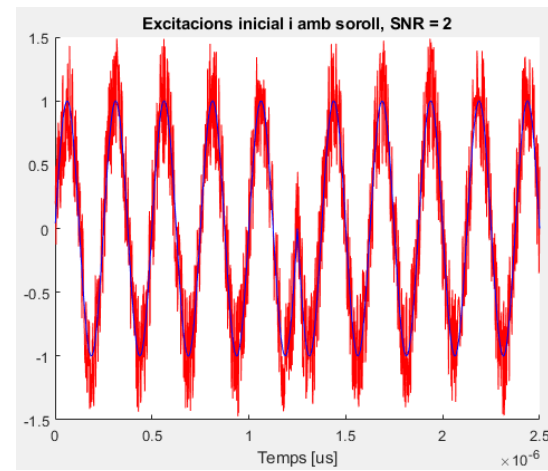
(a) Excitació abans (blau) i després (vermell) del soroll, SNR = 10



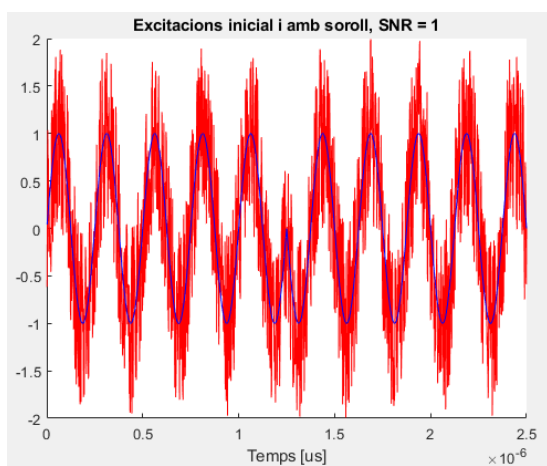
(b) Excitació abans (blau) i després (vermell) del soroll, SNR = 5



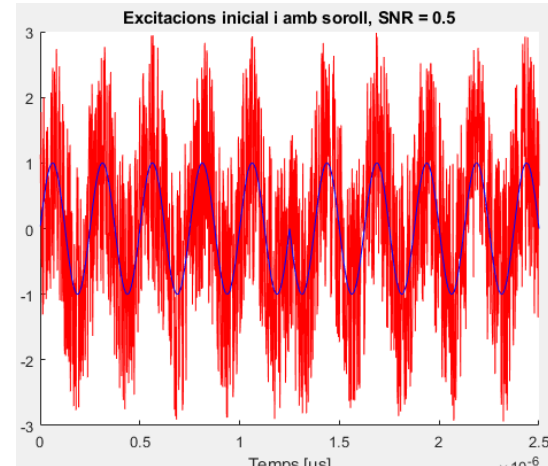
(c) Excitació abans (blau) i després (vermell) del soroll, SNR = 3



(d) Excitació abans (blau) i després (vermell) del soroll, SNR = 2

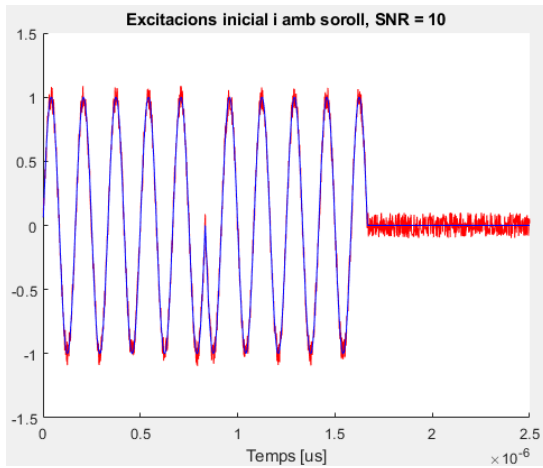


(e) Excitació abans (blau) i després (vermell) del soroll, SNR = 1

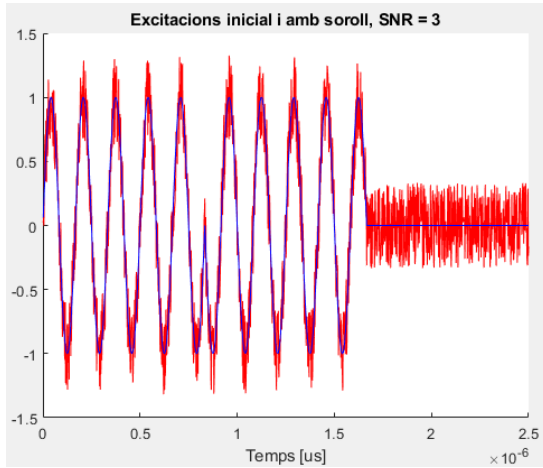
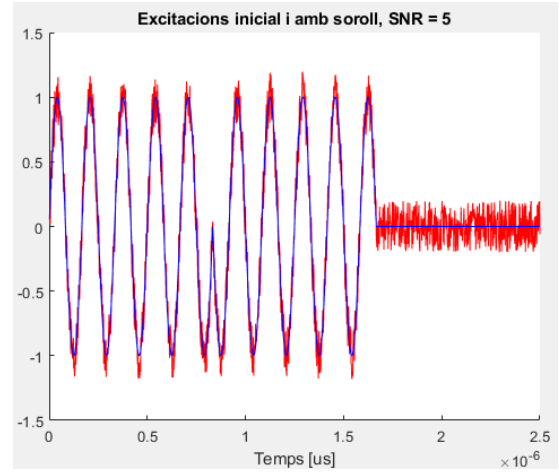


(f) Excitació abans (blau) i després (vermell) del soroll, SNR = 0.5

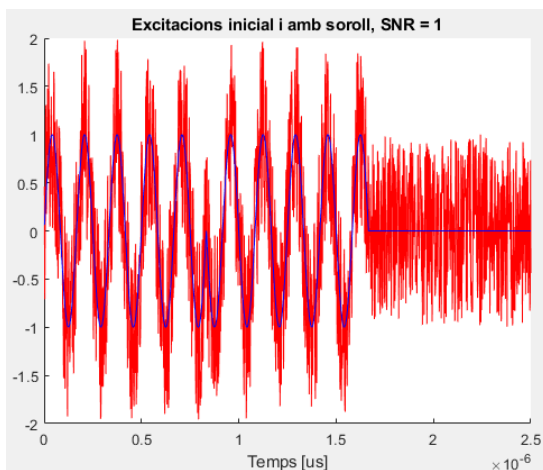
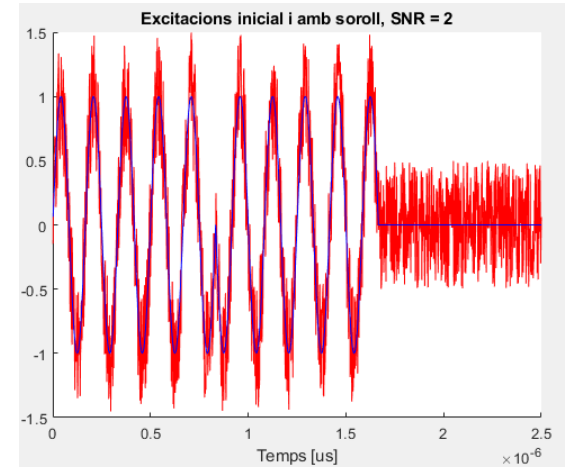
Figura 6.14: Excitacions abans (blau) i després (vermell) del soroll per diferents valors de SNR. Freqüència a 4 MHz



(a) Excitació abans (blau) i després (vermell) del so- (b) Excitació abans (blau) i després (vermell) del so-
roll, SNR = 10 roll, SNR = 5



(c) Excitació abans (blau) i després (vermell) del so- (d) Excitació abans (blau) i després (vermell) del so-
roll, SNR = 3 roll, SNR = 2



(e) Excitació abans (blau) i després (vermell) del so- (f) Excitació abans (blau) i després (vermell) del so-
roll, SNR = 1 roll, SNR = 0.5

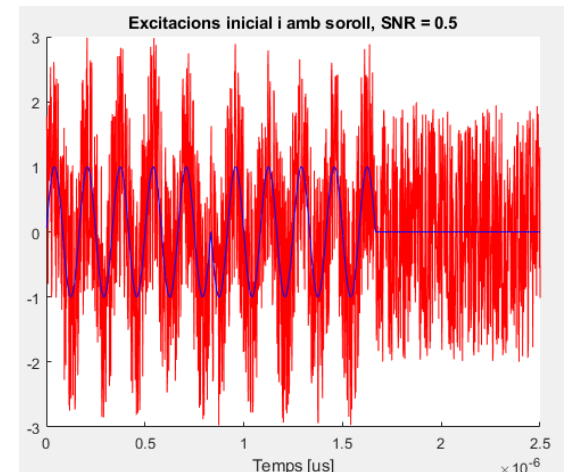


Figura 6.15: Excitacions abans (blau) i després (vermell) del soroll per diferents valors de SNR. Freqüència a 6 MHz

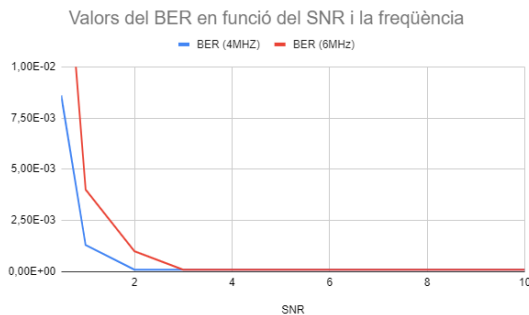
Donat que el soroll s'ha generat correctament, es realitza el càlcul del BER per cada un dels casos visualitzats.

Freqüència 4MHz				Freqüència 6MHz			
SNR	SNR (dB)	Nº bits	BER	SNR	SNR (dB)	Nº bits	BER
10	10	10000	$\leq 10^{-4}$	10	10	10000	$\leq 10^{-4}$
5	7	10000	$\leq 10^{-4}$	5	7	10000	$\leq 10^{-4}$
3	4.8	10000	$\leq 10^{-4}$	3	4.8	10000	$\leq 10^{-4}$
2	3	10000	$\leq 10^{-4}$	2	3	10000	$1 * 10^{-3}$
1	0	10000	$1.3 * 10^{-3}$	1	0	10000	$4 * 10^{-3}$
0.5	-3	10000	$8.6 * 10^{-3}$	0.5	-3	10000	$1.9 * 10^{-2}$

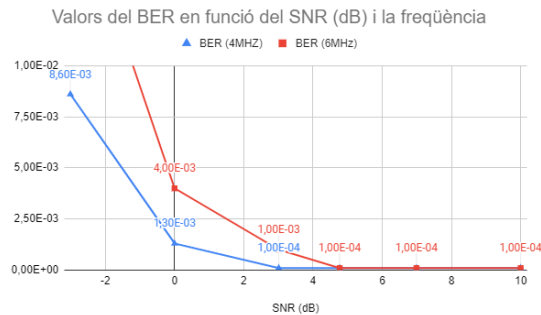
Taula 6.1: Obtenció dels diferents BER del sistema per cada SNR i freqüència.

Es genera la taula 6.1, que conté la mitjana de calcular els valors del BER 5 cops. Per raons de computació, no s'han pogut realitzar simulacions de més de 10000 bits donat que les simulacions amb aquest valor ja excedien la hora de treball de còmput.

Això implica que com a molt es pot detectar un BER mínim d'1bit/10000bits, que coincideix amb el BER suficient per una comunicació Full-Duplex en streaming, $BER \leq 10^{-4}[\text{Pop}+18]^2$.



(a) Valors del BER del sistema per cada freqüència i SNR.



(b) Valors del BER del sistema per cada freqüència i SNR (dB).

Segons aquest criteri, per valors de SNR per sobre de 3dB la transmissió a 4 MHz resulta satisfactòria. Tot i això, cal augmentar el SNR fins a 4.8dB o més per assolir els criteris de comunicació pels dos canals freqüencials, de manera que es pugui establir el sistema Full-Duplex desitjat, tal i com s'ha explicat a la figura 1.2.

Finalment hem assolit l'objectiu inicial, establint en 4.8dB el SNR mínim per treballar a les

²Flavius Pop et al. "Novel pMUT-Based Acoustic Duplexer for Underwater and Intrabody Communication". A.: Oct. de 2018, pàg. 1-4. DOI: 10.1109/ULTSYM.2018.8579709.

frequències tractades.

Conclusions

Com em temia al començar el projecte, l'impacte del COVID-19 ha sigut notori en el desenvolupament del mateix. El tancament de laboratoris m'ha obligat a modificar l'abast i la metodologia de la totalitat del treball donat que no pretenia usar simulacions, sinó un sistema real per tal d'obtenir valors empírics amb totes les consideracions necessàries.

Referent a la simulació, la combinació dels programes LTspice i Matlab m'ha comportat resultats molt satisfactoris, que es podrien incrementar tal i com s'ha mencionat a l'inici del capítol 6 automatitzant completament la interacció entre els programes en el moment de la simulació.

Degut a que he elaborat la totalitat del codi principal íntegrament per aquest treball (a excepció de les funcions *LTspice2Matlab* i del soroll gaussià *awgn*), és segur es pot modificar el codi per obtenir uns nivells d'eficiència i funcionalitat superiors. Aquest fenomen es pateix notablement quan es supera el nombre de mil bits per simulació, augmentant des de 3 minuts (1000 bits) fins a més d'1 hora (10.000 bits).

Tot i així he pogut assolir l'objectiu plantejat inicialment, trobar els límits de funcionament en funció del soroll d'un sistema de comunicacions Full-Duplex dissenyat i simulat en líquid on els transceptors estan constituïts per arrays de 10x10 PMUTs.

Referent a les freqüències centrals de treball (4 i 6 MHz), he observat que la freqüència baixa té una major immunitat al soroll (funciona en medis amb $\text{SNR} = 3\text{dB}$), però per definició també disposa d'una capacitat menor de transmissió de dades (800Kbits/s). Aquestes dades es contraposen amb la necessitat d'un $\text{SNR} = 4.8\text{dB}$ i una transmissió de 1200Kbits/s per la freqüència més elevada.

L'esmentada diferència freqüencial es pot aprofitar per aportar més ample de banda a la transmissió en el sentit que es requereixi.

Tal i com s'ha analitzat al capítol 5, s'assoleixen velocitats reals estimades de transferència compreses entre 600 i 900 Kbits/s en funció de la freqüència que s'observi. Això permet un nivell acceptable de transmissió de dades, missatges d'àudio i certs nivells de transmissió de vídeo.

Referències

- [KVP00] K.V.Puglia. *A General Design Procedure for Bandpass Filters Derived from Low Pass Prototype Elements: Part I*. Des. de 2000. URL: <https://www.microwavejournal.com/articles/3102-a-general-design-procedure-for-bandpass-filters-derived-from-low-pass-prototype-elements-part-i#:~:text=A%5C%20low%5C%20pass%5C%20filter%5C%20may%5C%20be%5C%20converted%5C%20to%2C,mapping%5C%20function%5C%20may%5C%20be%5C%20intuitively%5C%20or%5C%20mathematically%5C%20derived.> (cons. 22-08-2020).
- [Wag09] Paul Wagner. *Fast Import of Compressed Binary .RAW Files Created with LTspice Circuit Simulator*. Abr. de 2009. URL: <https://es.mathworks.com/matlabcentral/fileexchange/23394-fast-import-of-compressed-binary-raw-files-created-with-ltspice-circuit-simulator?focused=5113448&tab=function> (cons. 22-08-2020).
- [Wor12] RF Wireless World. *BPSK-Binary Phase Shift Keying*. 2012. URL: <https://www.rfwireless-world.com/Terminology/BPSK.html> (cons. 22-08-2020).
- [WSL15] Tao Wang, Renshi Sawada i Chengkuo Lee. “A Piezoelectric Micromachined Ultrasonic Transducer (pMUT) Using Piston-like Membrane Motion”. A: *Electron Device Letters, IEEE* 36 (set. de 2015), pàg. 957-959. DOI: 10.1109/LED.2015.2459075.
- [Gan+16] Adam Gannon et al. “Wireless Communications in Intra-Body Networks”. A: (2016).
- [Not16] Electronics Notes. *What is Bit Error Rate: BER tutorial*. 2016. URL: <https://www.electronics-notes.com/articles/radio/bit-error-rate-ber/what-is-ber-definition-tutorial.php> (cons. 22-08-2020).
- [SM16] G. Enrico Santagati i Tommaso Melodia. “Experimental Evaluation of Impulsive Ultrasonic Intra-Body Communications for Implantable Biomedical Devices”. A: *IEEE Transactions on Mobile Computing* 16 (gen. de 2016), pàg. 1-1. DOI: 10.1109/TMC.2016.2561277.
- [Yan16] Carmen Emily Yang. *What is the Piezoelectric Effect?* Set. de 2016. URL: <https://www.electronicdesign.com/power-management/article/21801833/what-is-the-piezoelectric-effect#:~:text=%5C%20What%5C%20is%5C%20the%5C%20Piezoelectric%5C%20Effect%5C%20%3F%5C%20%5C%201,inverse%5C%20piezoelectric%5C%20effect%5C%20sends%5C%20out%5C%20a...%5C%20More%5C%20> (cons. 22-08-2020).
- [Li+17] Junhong Li et al. “Design and Fabrication of Piezoelectric Micromachined Ultrasound Transducer (pMUT) with Partially-Etched ZnO Film”. A: *Sensors (Switzerland)* 17 (juny de 2017). DOI: 10.3390/s17061381.

- [Soc17] SocialGamers. *Tabla de Bitrates*. Gen. de 2017. URL: <https://www.socialgamers.es/tabla-bitrates-obs/> (cons. 21-08-2020).
- [Pop+18] Flavius Pop et al. "Novel pMUT-Based Acoustic Duplexer for Underwater and Intrabody Communication". A: oct. de 2018, pàg. 1-4. DOI: 10.1109/ULTSYM.2018.8579709.
- [Ele19] ElectronicsHub. *Butterworth Filter*. Febr. de 2019. URL: https://www.electronicshub.org/butterworth-filter/#%5C%EF%5C%BB%5C%BFButterworth_Approximations%5C%EF%5C%BB%5C%BF (cons. 22-08-2020).
- [Her+19] Bernard Herrera et al. "Frequency Agile PMUT-Based Ultrasonic Communication Links". A: gen. de 2019, pàg. 974-977. DOI: 10.1109/MEMSYS.2019.8870840.
- [Mat19] Aaron Mattmiller. *Butterworth Filter*. Gen. de 2019. URL: <https://medium.com/@amattmiller/running-ltspice-from-matlab-630d551032cc> (cons. 22-08-2020).
- [Led+20] Eyglis Ledesma et al. "Tent-plate AlN PMUT with a piston-like shape under liquid operation". A: *IEEE Sensors Journal* PP (maig de 2020), pàg. 1-1. DOI: 10.1109/JSEN.2020.2995643.
- [Pop+20] Flavius Pop et al. "Enabling Real-Time Monitoring of Intrabody Networks through the Acoustic Discovery Architecture". A: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* PP (juny de 2020), pàg. 1-1. DOI: 10.1109/TUFFC.2020.3002973.
- [Sol20] Cadence PCB Solutions. *What is Signal to Noise Ratio and How to calculate it?* Gen. de 2020. URL: <https://resources.pcb.cadence.com/blog/2020-what-is-signal-to-noise-ratio-and-how-to-calculate-it> (cons. 22-08-2020).
- [Cor] Midé Technology Engineering Corporation. *The History of Piezoelectricity*. URL: <https://piezo.com/pages/history-of-piezoelectricity> (cons. 28-08-2020).

Referències a articles

- [WSL15] Tao Wang, Renshi Sawada i Chengkuo Lee. "A Piezoelectric Micromachined Ultrasonic Transducer (pMUT) Using Piston-like Membrane Motion". A: *Electron Device Letters, IEEE* 36 (set. de 2015), pàg. 957-959. DOI: 10.1109/LED.2015.2459075.
- [Gan+16] Adam Gannon et al. "Wireless Communications in Intra-Body Networks". A: (2016).
- [SM16] G. Enrico Santagati i Tommaso Melodia. "Experimental Evaluation of Impulsive Ultrasonic Intra-Body Communications for Implantable Biomedical Devices". A: *IEEE Transactions on Mobile Computing* 16 (gen. de 2016), pàg. 1-1. DOI: 10.1109/TMC.2016.2561277.
- [Li+17] Junhong Li et al. "Design and Fabrication of Piezoelectric Micromachined Ultrasound Transducer (pMUT) with Partially-Etched ZnO Film". A: *Sensors (Switzerland)* 17 (juny de 2017). DOI: 10.3390/s17061381.
- [Led+20] Eyglis Ledesma et al. "Tent-plate AlN PMUT with a piston-like shape under liquid operation". A: *IEEE Sensors Journal* PP (maig de 2020), pàg. 1-1. DOI: 10.1109/JSEN.2020.2995643.

- [Pop+20] Flavius Pop et al. “Enabling Real-Time Monitoring of Intrabody Networks through the Acoustic Discovery Architecture”. A: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* PP (juny de 2020), pàg. 1-1. DOI: 10.1109/TUFFC.2020.3002973.

Referències a procediments

- [Pop+18] Flavius Pop et al. “Novel pMUT-Based Acoustic Duplexer for Underwater and Intrabody Communication”. A: oct. de 2018, pàg. 1-4. DOI: 10.1109/ULTSYM.2018.8579709.
- [Her+19] Bernard Herrera et al. “Frequency Agile PMUT-Based Ultrasonic Communication Links”. A: gen. de 2019, pàg. 974-977. DOI: 10.1109/MEMSYS.2019.8870840.

Referències a pàgines web

- [KVP00] K.V.Puglia. *A General Design Procedure for Bandpass Filters Derived from Low Pass Prototype Elements: Part I*. Des. de 2000. URL: <https://www.microwavejournal.com/articles/3102-a-general-design-procedure-for-bandpass-filters-derived-from-low-pass-prototype-elements-part-i#:~:text=A%5C%20low%5C%20pass%5C%20filter%5C%20may%5C%20be%5C%20converted%5C%20to%2C%20mapping%5C%20function%5C%20may%5C%20be%5C%20intuitively%5C%20or%5C%20mathematically%5C%20derived.> (cons. 22-08-2020).
- [Wag09] Paul Wagner. *Fast Import of Compressed Binary .RAW Files Created with LTspice Circuit Simulator*. Abr. de 2009. URL: <https://es.mathworks.com/matlabcentral/fileexchange/23394-fast-import-of-compressed-binary-raw-files-created-with-ltspice-circuit-simulator?focused=5113448&tab=function> (cons. 22-08-2020).
- [Wor12] RF Wireless World. *BPSK-Binary Phase Shift Keying*. 2012. URL: <https://www.rfwireless-world.com/Terminology/BPSK.html> (cons. 22-08-2020).
- [Not16] Electronics Notes. *What is Bit Error Rate: BER tutorial*. 2016. URL: <https://www.electronics-notes.com/articles/radio/bit-error-rate-ber/what-is-ber-definition-tutorial.php> (cons. 22-08-2020).
- [Yan16] Carmen Emily Yang. *What is the Piezoelectric Effect?* Set. de 2016. URL: <https://www.electronicdesign.com/power-management/article/21801833/what-is-the-piezoelectric-effect#:~:text=%5C%20What%5C%20is%5C%20the%5C%20Piezoelectric%5C%20Effect%5C%3F%5C%20%5C%201%2C%20inverse%5C%20piezoelectric%5C%20effect%5C%20sends%5C%20out%5C%20a...%5C%20More%5C%20> (cons. 22-08-2020).
- [Soc17] SocialGamers. *Tabla de Bitrates*. Gen. de 2017. URL: <https://www.socialgamers.es/tabla-bitrates-obs/> (cons. 21-08-2020).
- [Ele19] Electronicshub. *Butterworth Filter*. Febr. de 2019. URL: https://www.electronicshub.org/butterworth-filter/#%5C%EF%5C%BB%5C%BFButterworth_Approximations%5C%EF%5C%BB%5C%BF (cons. 22-08-2020).

- [Mat19] Aaron Mattmiller. *Butterworth Filter*. Gen. de 2019. URL: <https://medium.com/@amattmiller/running-ltspice-from-matlab-630d551032cc> (cons. 22-08-2020).
- [Sol20] Cadence PCB Solutions. *What is Signal to Noise Ratio and How to calculate it?* Gen. de 2020. URL: <https://resources.pcb.cadence.com/blog/2020-what-is-signal-to-noise-ratio-and-how-to-calculate-it> (cons. 22-08-2020).
- [Cor] Midé Technology Engineering Corporation. *The History of Piezoelectricity*. URL: <https://piezo.com/pages/history-of-piezoelectricity> (cons. 28-08-2020).